

ABSTRACT

KRIPAKARAN, PRAKASH. GA-Based Decision Support for Optimizing the Response of Secondary Systems. (Under the direction of Dr. Abhinav Gupta).

The objective of this research is to develop a Decision Support System (DSS) for seismic design and performance evaluation of piping supports. The current practice of designing piping support locations is primarily heuristic, relying heavily on professional experience. In this thesis, approaches to optimize support locations using Genetic Algorithms (GAs), a heuristic optimization technique, are discussed. These approaches have been implemented in a DSS using Vitri, a generic, distributed framework designed to support the development of DSSs, which reduces the computational requirements by combining the processing power of a network of workstations.

Previous attempts to solve the problem of pipe support optimization modeled supports as flexible springs, which have a stiffness depending on the support capacity, resulting in the use of an integer representation in the GA. In this thesis, a new approach where supports are modeled as rigid springs is presented. This permits the use of a binary representation in the GAs. Also, earlier attempts had solved the problem by minimizing the number of supports, which does not always indicate if cost is minimized. In this thesis, capital cost and lifetime cost are studied by examining the trade-off curve between the cost and the number of supports. A crossover scheme aimed at generating cost optimal solutions of a specified number of supports, which is required for generating trade-off curves, is proposed.

It has been observed that optimization results in solutions that may be practically infeasible because of unmodeled costs in the optimization model. In pipe support optimization, such costs might be from the preference of certain locations over others because of easier support installation costs or the desire to locate the supports under lumped masses to stabilize the pipe against local vibrations from equipment such as pumps and motors. The role of Modeling to Generate Alternatives (MGA), a methodology based on optimization to produce alternatives, is explored to address these issues.

**GA-Based Decision Support for Optimizing the
Response of Secondary Systems**

by

Prakash Kripakaran

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Department of Civil Engineering

Raleigh

2002

Approved By:

Dr. John W. Baugh, Jr.

Dr. G. (Kumar) Mahinthakumar

Dr. Abhinav Gupta
Chair of Advisory Committee

Biography

Prakash Kripakaran, was born on June, 16, 1979, in Coimbatore, India, where he did most of his schooling. His family moved to Madras in May 1993, where he went to high school. He completed his Bachelor's degree in Civil engineering at Indian Institute of Technology, Madras. He joined the Masters program in Civil engineering in North Carolina State University in the fall of 2000 and has been working on this project from then on. He plans to continue his research in this subject by pursuing a doctorate in Civil engineering at North Carolina State University.

Acknowledgements

First of all, I would like to thank Dr. Abhinav Gupta for his guidance and encouragement throughout the course for this research. I would also like to thank the other members of my committee, Dr. Baugh and Dr. Kumar, who along with my advisor Dr. Gupta, made themselves available for a meeting almost every week, to discuss the progress of the research. The discussion and the ideas generated during the meeting went a long way in furthering this research.

I would also like to thank my friend, Sujay Kumar, who was a graduate student when I joined the Masters program. He helped me learn Vitri, a framework he had developed during his doctoral research at North Carolina State University, which I used all through my research.

Lastly, this research would not have been possible without the moral support of my parents.

Table of Contents

List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 General background	1
1.2 Objectives	3
1.3 Organization	4
2 The Decision Support System	6
2.1 Design of DSS	8
2.2 Vitri: A framework for engineering DSS	9
3 Optimizing Seismic Response of Secondary Systems	12
3.1 Introduction	12
3.2 Design of piping supports for seismic loads	14
3.3 GA-based pipe support optimization	17
3.4 Cost optimization	22
3.4.1 Example piping system	23
3.4.2 Capital cost optimization	24
3.4.3 Lifetime cost optimization	26
3.5 Modified objective function	27
3.6 Modeling to Generate Alternatives (MGA)	28
3.7 Application of MGA to pipe support optimization	31
3.8 Application to real life piping system	33
3.9 Summary and conclusions	34
4 Other Results	57
4.1 Integer string representation	57
4.2 Cumulative capacity optimization	58

4.3	Comparison of selection techniques	58
4.4	Uniform crossover	59
4.4.1	Lifetime cost optimization	59
4.4.2	Modified fitness function	60
4.4.3	MGA solutions	61
4.5	Application to real life piping system	61
5	Summary and Conclusions	71
5.1	Summary	71
5.2	Conclusions	73
5.3	Future work	74
	References	76

List of Figures

2.1	A sample graphical interface offered by Vitri	11
3.1	Straight pipe with lumped masses	41
3.2	Interfacing of PIPESTRESS and Vitri	42
3.3	Illustration of the proposed crossover	43
3.4	Results of multi-objective study for capital cost	44
3.5	Results of multi-objective study for lifetime cost	44
3.6	Optimal solutions for capital cost obtained using proposed crossover	45
3.7	Optimal solutions for lifetime cost obtained using proposed crossover	46
3.8	Optimal solutions for lifetime cost obtained using proposed crossover	47
3.9	θ_k/θ_{kc} for supports of solutions in Figure 3.6	48
3.10	θ_k/θ_{kc} for supports of solutions in Figure 3.7	48
3.11	Best fitness solutions using modified fitness function (Equation 3.14) for lifetime cost	49
3.12	Multiple-objective optimization	50
3.13	Decision space for LP example	51
3.14	Flowchart showing incorporation of MGA in Vitri	52
3.15	MGA solutions for lifetime cost with $n_{req} = 5$	53
3.16	MGA solutions for lifetime cost with $n_{req} = 7$	54
3.17	Real piping system (not to scale)	55
3.18	Multi-objective study of lifetime cost for the real piping system	56
4.1	Comparison of performance of various crossover schemes for lifetime cost	63
4.2	Multi-objective study cumulative capacity as cost	64
4.3	Comparison of selection schemes for capital cost using multiple seeds in GA	64
4.4	Comparison of selection schemes for lifetime cost using multiple seeds in GA	65
4.5	Optimal lifetime cost solutions obtained using uniform crossover and bi- nary representation	66

4.6	Optimal capital cost solutions obtained using uniform crossover and binary representation	67
4.7	θ_k/θ_{kc} values for solutions given in Figure 4.5	68
4.8	θ_k/θ_{kc} values for solutions given in Figure 4.6	68
4.9	The best solution and the alternatives using Equation 4.2	69
4.10	Alternatives generated for lifetime cost optimal solution of $n_{req} = 8$	70

List of Tables

3.1	Location of lumped masses along the pipe	36
3.2	Cost and capacity of supports	37
3.3	Properties of optimal capital cost solutions given in Figure 3.6	37
3.4	Anchor forces of optimal capital cost solutions given in Figure 3.6	38
3.5	Properties of optimal lifetime cost solutions given in Figures 3.7 and 3.8	38
3.6	Anchor forces of optimal lifetime cost solutions given in Figures 3.7 and 3.8	39
3.7	Properties of optimal solutions obtained using Equation 3.14	39
3.8	Properties of MGA solutions for $n_{req} = 5$ in Figure 3.15	39
3.9	Anchor forces of MGA solutions for $n_{req} = 5$ in Figure 3.15	40
3.10	Properties of MGA solutions for $n_{req} = 7$ in Figure 3.16	40
3.11	Anchor forces of MGA solutions for $n_{req} = 7$ in Figure 3.16	40
4.1	Properties of solutions given in Figure 4.9	63
4.2	Properties of solutions given in Figure 4.10	63

Chapter 1

Introduction

1.1 General background

Seismic design and qualification of secondary systems such as power plant piping, like other complex decision problems, involves many tradeoffs among competing criteria. At all stages in a power plant life-cycle, structural engineers are faced with formidable challenges of complex decision-making. In the context of seismic design and performance evaluation, these may relate to technical as well as practical feasibility, cost, and redundancy. Although some of these challenges can be quantified and formally analyzed, others may call for outside opinions from experts or engineering judgment. Key features of the decision-making process include identification of alternatives for solving the given problem. Then, whether an alternative is satisfactory typically requires analysis by one or more computer models, e.g., redundancy versus cost analyses, and the expertise and judgment of the engineer.

Research in structural optimization has included computation complexity studies [21], traditional mathematical optimization [9], and modern heuristic optimization such as genetic algorithms [20]. In practical applications, the use of mathematical optimization or other formal search techniques may lead to “optimal” structural systems that are either infeasible or unnecessarily costly in the sense that they are not conducive to construction, maintenance or serviceability. In this scenario, the optimization model produces a single solution that is “best,” but only with respect to the objectives and constraints that appear in the model: the unmodeled issues, such as downstream maintenance or outage costs, may not have been factored into the search process. To produce a truly optimal solution, then, would require that the judgment and expertise of an experienced designer somehow be embedded in the model. These difficulties in modeling are a key limitation, not of optimization techniques per se, but of the way in which they are routinely used.

Optimization for seismic support locations in piping systems is a highly complex and iterative problem that is time and cost intensive. An extremely large number of possibilities may exist, making enumeration impractical. The current practice of designing piping support locations is primarily heuristic, relying heavily on professional experience. Engineers use the knowledge learned from observed performances and past failures to arrive at the most appropriate locations. In this paper, we explore the role of formal computational approaches that support decision making in the seismic design and performance evaluation of multiply-supported piping systems in critical industrial facilities such as nuclear power plants.

1.2 Objectives

A primary system (such as building) is a structural system that receives seismic input directly from the ground and filters it to the secondary system supported on it. Secondary systems (such as equipment and piping) are supported on primary systems and do not receive seismic input directly from the ground. Seismic design for piping is dependent on the seismic input it receives from the primary system (building). The piping receives this input through the supports on which it is mounted. The seismic input transmitted through all the supports may not be the same. It typically depends upon the floor elevations of the supports in the primary system. Choosing appropriate support locations for the pipe optimizes the overall cost of the supports and the response. The existence of numerous possibilities in selecting the support locations and the absence of a strategy for choosing the support locations make it a complex-decision making problem. The primary objective of this study is to explore the role of a Decision Support System (DSS) in seismic design and performance evaluation of piping supports. This is realized using the following components:

- PIPESTRESS [8] - a finite element analysis software package, for evaluating the seismic response of piping systems.
- Vitri [14] - a distributed, high performance computing framework designed to support Decision Support Systems.
- Heuristic techniques such as Genetic Algorithms [10, 17] for pipe support optimization.

- MGA [4] - “Modeling to Generate Alternatives,” a methodology that utilizes optimization, to generate alternatives that are similar in objective space, but very different in decision space.

1.3 Organization

The thesis consists of five chapters. This chapter gives the introduction, objectives and the overall organization. The second chapter discusses the concepts of the DSS to be designed for the given problem. The DSS has been implemented on Vitri, a distributed object-oriented framework developed for DSSs. The tools provided by Vitri and PIPESTRESS, a commercial package for finite element analysis, are presented. The purpose of the DSS and the approach taken to develop the DSS are described in detail.

The third chapter presents a GA-based optimization approach for choosing the seismic support locations for power plant piping. It discusses improvements to various components of the GA, such as, representation and crossover. It explains the optimization procedure used for multi-objective study between cost (capital and lifetime cost) and the number of supports and examines the trade-off curves obtained. It describes the MGA technique used to generate alternatives to the optimal solutions obtained from the GA approach and analyzes the generated results. Finally, the developed GA approach is applied to a real life piping system and the results of the multi-objective study between lifetime cost and the number of supports are given. It should be noted that parts of the introduction and the

objectives are repeated in chapter 3. The third chapter is essentially a manuscript to be submitted to the ASCE Journal of Structural Engineering.

The fourth chapter consists of the results generated during the course of research but are not given in the third chapter representing the manuscript. These include (a) comparison of the performance of integer string representation with binary string representation, (b) the alternatives generated by MGA technique while using uniform crossover, (c) results on the selection techniques studied for lifetime cost optimization, and (e) intermediate results obtained on application to real-life piping system. The last chapter gives the conclusions and summary.

Chapter 2

The Decision Support System

Decision support systems (DSSs) are formal approaches for computer assisted decision making. A DSS is a computer system that initiates human involvement by helping engineers to create alternatives, test their effects and interpret the generated results. A DSS uses tools for analysis, optimization, etc., to assist the engineer in evaluating the alternatives. Thus, engineers can select favorable alternatives for implementation.

Often, the decision maker does not have a full understanding of the problem and its parameters at the onset of the decision-making process. He strives to get a better understanding of the problem through experimentation and trial-and-error. A decision making process can involve (a) tasks that require computer-intensive processing and (b) tasks that require human judgment. A computer-based DSS [15, 18, 12] enhances the decision-making process by providing the engineer with an easy-to-use problem solving environment that supports a suite of analysis tools for joint-cognitive decision making. A joint-cognitive system

is comprised of a human decision maker and a set of computational resources. DSSs allow the engineer to monitor progress by presenting him with intermediate results for review. It supports a collection of various analytical and computation techniques, schemes for uncertainty propagation, and optimization tools.

The development of modern heuristic techniques such as genetic algorithms, combined with high performance computing environments, have increased the scope of engineering design problems that can be addressed by mathematical optimization. New approaches for decision support where the intended role of optimization models is not to generate the “best” solution, but rather to generate sets of good, alternative designs that provide insight and creativity, have been developed. This approach has been applied for alternative generation in the design of truss structures using integer-linear programming [1], vehicle routing and scheduling using simulated annealing [2] and air quality management for control-strategy optimization using genetic algorithms and simulated annealing [16], and have been shown to greatly assist decision-making.

A distributed computing system can be used to improve the performance of computationally intensive DSS problems. For instance, computational resources can be assigned jobs such as brute force calculations and information storage and retrieval. This permits human decision makers to set their minds on activities such as consideration of abstract issues, directing overall decision-making process, and providing feedback based on intuition and professional judgment.

The decision maker selects the parameters, schemes and software, and decides the de-

sign and serviceability requirements for compliance. These selections are then explored with the help of the DSS. The decision maker uses his professional judgment for providing useful feedback to the DSS in the form of changes made to the previous selections. In this manner, the DSS employs an iterative process involving continuous modification to evolve initially conceived potential solutions to a satisfactory solution. The following section describes the design of the DSS for seismic design and performance evaluation of secondary systems.

2.1 Design of DSS

The DSS draws on the complementary strengths of the engineer and the computer in a joint-cognitive system. Three components comprise the DSS,

- Finite element analyses are used for evaluating the seismic response of piping systems.
- Genetic algorithms are used as part of a formal procedure for generating alternative solutions using a methodology referred to as “modeling to generate alternatives - MGA” [4]. MGA uses optimization to generate a set of solutions that are similar in objective space - “near optimal,” but very different in decision space - “as different as possible.”
- The computational performance requirements of the DSS are addressed by using an existing framework for high performance computing on workstation clusters. The

object-oriented framework, Vitri, includes basic support for distributed computing and communication, and distributed GA implementations [3].

2.2 Vitri: A framework for engineering DSS

Vitri [14] is a generic, distributed, object-oriented framework for the development of engineering DSSs. Vitri provides a high performance computing environment for distributed computing that harnesses the resources of a network of computers to provide adequate computing power to deal with the various problems. A lot of formal tools and concepts such as optimization-based techniques have been developed, that help in searching for solutions that optimize certain goals while satisfying specified constraints. The development of such tools has helped in avoiding time-consuming enumeration of potential solutions for problems where good solutions are hard to find using a trial-and-error approach. Vitri provides a number of formal computational approaches such as heuristic optimization techniques that provide generic capabilities that are suitable for dealing with difficult, ill-behaved problems.

A decision maker is typically interested in details such as the trade-offs between cost and a certain competing variable, the effect of uncertainties on solutions, etc. Formal tools and concepts provided by Vitri can be used in an iterative decision making process to assist with such what-if analyses. Moreover, tools for optimization can be used to generate good alternatives in addition to finding optimal solutions. For example, Vitri provides a

tool, Modeling to Generate Alternatives (MGA) [4], that is based on genetic algorithms, to produce a small set of slightly sub-optimal solutions that are different in decision space.

Vitri also provides a number of graphical interfaces to evaluate and monitor the runtime performance of the distributed system. Decision makers can use these graphical interfaces to decide on whether to proceed with the exploration of an alternative. For instance, convergence towards an optimal solution for a computationally intensive technique may be aborted or re-routed by the decision maker based on the observations during the first few iterations. Vitri has a graphical display displaying the best fitness and the average fitness of the population as the GA progresses. A screen capture to illustrate this is given in Figure 2.1.

Engineers often develop prototypes based on their ideas, and test them to gain a better understanding. The framework in Vitri allows for rapid prototyping, which is desirable for the DSS as designers can learn about the problem quickly. The object-oriented tools and component-based modeling technologies of Vitri promote the development of modular systems that are flexible and extensible. This also means that the tools in Vitri can be reused for testing different prototypes and applications. Vitri not only facilitates an iterative decision making process, but also helps the decision maker incrementally learn about the problem at hand. The improved knowledge promotes the refinement of existing designs and models. The tools provided by Vitri to assist problem solving, and to insulate the user from the complexities of underlying hardware and software, make Vitri a powerful framework for the development of engineering DSSs.

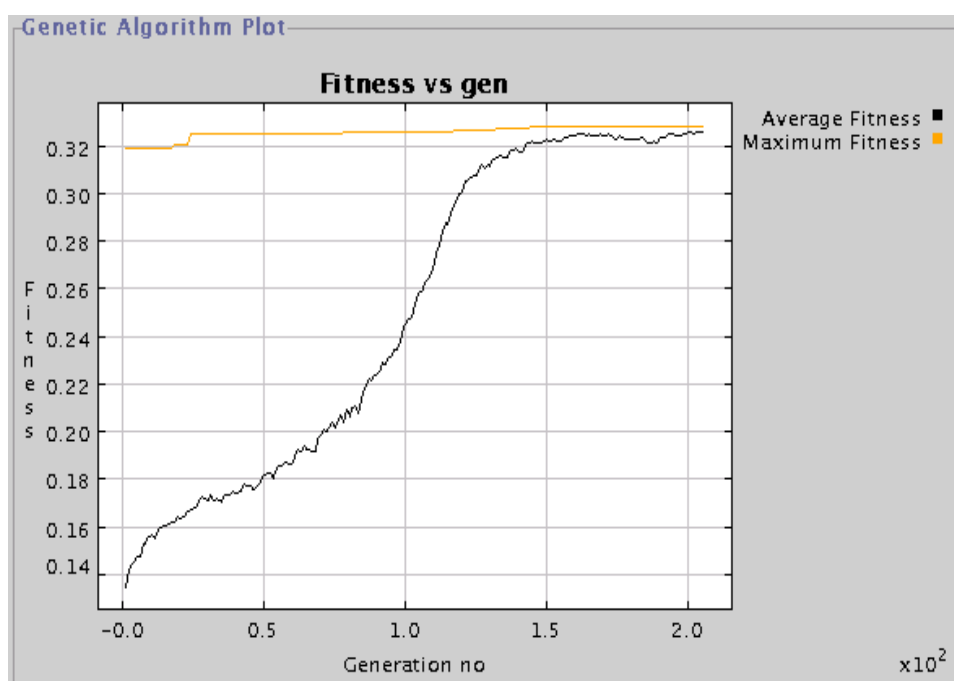


Figure 2.1: A sample graphical interface offered by Vitri

Chapter 3

Optimizing Seismic Response of Secondary Systems

3.1 Introduction

Seismic design and qualification of secondary systems such as power plant piping, like other complex decision problems, involves many tradeoffs among competing criteria. At all stages in a power plant life-cycle, structural engineers are faced with formidable challenges of complex decision-making. In the context of seismic design and performance evaluation, these may relate to technical as well as practical feasibility, cost, and redundancy. Although some of these challenges can be quantified and formally analyzed, others may call for outside opinions from experts or engineering judgment. Key features of the decision-making process include identification of alternatives for solving the given problem. Then, whether

an alternative is satisfactory typically requires analysis by one or more computer models, e.g., redundancy versus cost analyses, and the expertise and judgment of the engineer.

Research in structural optimization has included computation complexity studies [21], traditional mathematical optimization [9], and modern heuristic optimization such as genetic algorithms [20]. In practical applications, the use of mathematical optimization or other formal search techniques may lead to “optimal” structural systems that are either infeasible or unnecessarily costly in the sense that they are not conducive to construction, maintenance or serviceability. In this scenario, the optimization model produces a single solution that is “best,” but only with respect to the objectives and constraints that appear in the model: the unmodeled issues, such as downstream maintenance or outage costs, may not have been factored into the search process. To produce a truly optimal solution, then, would require that the judgment and expertise of an experienced designer somehow be embedded in the model. These difficulties in modeling are a key limitation, not of optimization techniques per se, but of the way in which they are routinely used.

Optimization for seismic support locations in piping systems is a highly complex and iterative problem that is time and cost intensive. An extremely large number of possibilities may exist, making enumeration impractical. The current practice of designing piping support locations is primarily heuristic, relying heavily on professional experience. Engineers use the knowledge learned from observed performances and past failures to arrive at the most appropriate locations. In this paper, we explore the role of formal computational approaches that support decision making in the seismic design and performance evaluation

of multiply-supported piping systems in critical industrial facilities such as nuclear power plants.

The objectives of the present study are realized through a prototype decision support system (DSS) that draws on the complementary strengths of the engineer and the computer in a joint-cognitive system. Three components comprise the DSS. First, finite element analyses are used for evaluating the seismic response of piping systems. Second, genetic algorithms are used as part of a formal procedure for generating alternative solutions using a methodology referred to as “modeling to generate alternatives - MGA” [4]. MGA uses optimization to generate a set of solutions that are similar in objective space - “near optimal,” but very different in decision space - “as different as possible.” Third, the computational performance requirements of the DSS are addressed by using an existing framework for high performance computing on workstation clusters. The object-oriented framework, Vitri, includes basic support for distributed computing and communication, and distributed GA implementations [3].

3.2 Design of piping supports for seismic loads

Conventionally, an initial design of the piping system and its supports is drawn primarily from the standpoint of withstanding loads calculated from thermal, hydraulic, and deadweight considerations. In piping systems that experience significant thermal stresses (hot piping), additional supports are likely to increase the thermal stresses due to increased

constraints. At the same time, additional supports are needed to withstand earthquake loads. Supports called snubbers are used to satisfy these competing requirements. Snubbers have a locking mechanism that activates only during pipe vibrations, thereby avoiding additional constraints against thermal loads in normal operation. Snubbers can be either hydraulic or mechanical devices that require periodic maintenance. Regulations dictate that the integrity of critical piping systems and pipe supports be maintained at all times to withstand future earthquakes safely. It should be noted that pipe vibrations due to hydraulic loads such as water hammer cause snubbers to lock and activate during normal operation. Emergency plant outages arise whenever a snubber is found to have failed or malfunctioned. In addition to having high capital cost, the lifetime cost of snubbers is excessively high especially when the cost of replacement and loss of revenue during an outage are considered together with the maintenance cost. In piping systems that have negligible thermal stresses (cold piping), a designer can avoid the use of snubbers for withstanding seismic loads by using rigid or hanger type supports. While the capital and maintenance cost of such supports is significantly less than that of snubbers, the other issues related to lifetime cost and downtime remain the same.

The key elements of piping system design for withstanding seismic loads are determining the number of supports, the type of supports (capacities), and their locations. Some of the considerations in this process are: (1) use of as few supports as possible, which minimizes capital cost, (2) minimization of high capacity (and also high cost) supports to reduce the maintenance and replacement costs, (3) selection based on analysis for (a) identification

and elimination of cases with overstressed piping, (b) evaluating support loads and their acceptability against support capacities, and (c) evaluating maximum pipe displacements and their acceptability with respect to serviceability limits. An additional consideration that is not quantified in present practice is related to the desire for providing a support directly under the heavy equipment such as valves that may be located on the piping system. Further, a piping system that is very flexible is undesirable with respect to vibrations encountered during everyday operations, i.e., it is desired that the fundamental frequency of vibration be above 1 Hz. Typical values for piping system frequencies exceed 5 Hz. Additional constraints may also exist depending upon the particular problem or a given solution.

Chiba et al. [7] use genetic algorithms (GAs) for optimization in seismic design of pipe support locations and capacities. In this study, supports are modeled as uniaxial springs of varying stiffness. The objective function is formulated as:

$$\min\left\{\max_{0 \leq i \leq n_{elm}} \{RMS(\sigma_i)\} + \alpha n_{sup} + \beta \sum_{j=0}^{n_{sup}} k_j + \gamma P_s(\sigma) + \delta P_r(F)\right\} \quad (3.1)$$

where $RMS(\sigma_i)$ is the root mean square stress at location i , n_{elm} is the total number of elements used in the finite element model, n_{sup} is the total number of support locations, and k_j is the spring stiffness of the j^{th} support. α , β , γ , and δ are multipliers assigned values according to the relative importance of $RMS(\sigma_i)$, n_{sup} , $\sum k_j$, P_s and P_r . P_s and P_r are penalty functions to solutions in which the pipe stresses or support loads exceed the allowables, respectively. The above formulation is focused on minimizing (a) stresses in the piping system, (b) the number of supports, and (c) the cumulative support capacities that are represented in terms of spring stiffness. Minimization of the number of supports and

their cumulative capacities is a widely accepted practice that is implicitly targeted towards minimizing the cost associated with the supports. It should be noted that the above formulation does not consider any constraints on the displacement serviceability limits. Also, the modeling of supports as springs with varying stiffness, while an acceptable practice, can lead to infeasible solutions that may go undetected. For example, a piping analysis may yield axial displacements in supports that are represented by relatively soft springs. Such axial support displacements are usually incorrect because a typical support is axially rigid and fails in buckling at loads corresponding to their capacities. The objective of the study conducted by Chiba et al. [7] is to arrive at the most optimal solution with respect to the modeled issues. However, such a solution may be infeasible with respect to the other constraints that are not included in the model or the issues that cannot be quantified. For example, the desire to provide a support directly under a heavy equipment is not addressed in the optimal solution generated using Equation 3.1. Further, Equation 3.1 may not be acceptable with respect to typical piping failures that occur primarily due to support buckling and not due to excessive stresses.

3.3 GA-based pipe support optimization

For inherently discrete problems such as pipe support optimization, methods such as GAs are superior to traditional gradient-based techniques which require that the objective function be continuous with respect to decision variables. In the application of GAs to pipe

support optimization, the problem formulation requires an appropriate representation. The representation determines the size of the search space for the GA. Two different representations are possible depending upon the manner in which support topology and sizing are considered in the analysis model.

1. *Integer string*: An integer string representation is required if supports are modeled as flexible springs with specified stiffness values (corresponding to the support capacities). According to this representation, the length of GA string corresponds to the number of possible support locations such that each gene is represented by an integer that is used to identify the support type; i.e., a string of length n , $\langle b_1, b_2, \dots, b_n \rangle$ and

$$b_i = j, \quad 0 \leq j \leq m \quad (3.2)$$

where j is an integer used for identifying support type at location i and m is the number of support types. The search space size in this representation is $(m + 1)^n$. It can be observed that the size of the search space increases by a factor of n with an increase in the number of available support types.

2. *Binary string*: As described earlier, it may be desirable to model supports as rigid (a single spring having high stiffness value). Consequently, the primary decision in the GA is whether or not to have a support at a particular location i . A binary string may

then be used for the purpose such that

$$b_i = \begin{cases} 1, & \text{if a support exists at location } i \\ 0, & \text{if no support exists at location } i \end{cases}$$

Support types can be selected in each evaluation based on the support loads calculated from the analysis. The search space size when using this representation is much less and equal to only 2^n . Therefore, the GA not only converges faster but also produces better solutions due to a reduction in the search space. All the results and discussion that follow assume the use of a binary string representation.

Seeding of a GA can be critical with respect to efficiently searching the entire decision space without significantly compromising the time to convergence. For an integer string representation, a typical rolling of dice to seed the GA would result in a string whose genes have been assigned a random integer corresponding to a support type. However, this approach is undesirable as it results in seeding the population with strings that have supports at almost all possible locations. As designers are typically interested in studying the trade-off between cost and the number of supports, cost optimal solutions are needed for a specified number of supports n_{req} . Therefore, the GA can be seeded with solutions having nearly n_{req} supports if the value of b_i is selected using a random variable p which is uniformly distributed between 0 and 1.

$$b_i = \begin{cases} 0, & \text{if } p < \frac{n_{req}}{n} \\ j, & \text{if } p > \frac{n_{req}}{n} \text{ where } 1 \leq j \leq m \end{cases} \quad (3.3)$$

where n is the total number of possible locations. The value of j is selected by rolling a dice of value m . In a binary string representation, b_i can take on only two values, 0 or 1. A conventional seeding would result in nearly half as many support locations as the number of nodes in the piping model. Hence, b_i is chosen as follows:

$$b_i = \begin{cases} 0, & \text{if } p < \frac{n_{req}}{n} \\ 1, & \text{if } p > \frac{n_{req}}{n} \end{cases} \quad (3.4)$$

A better approach to obtain cost optimal solutions for a specified number of supports, n_{req} , would be to seed the genetic algorithm with solutions having exactly n_{req} number of supports. This can be achieved by choosing b_i as follows:

$$b_c = 1, \forall c \in C = \{c_1, c_2, \dots, c_{n_{req}}\} \quad (3.5)$$

where c_i are randomly chosen from the set of n available nodes and $c_i \neq c_j$ if $i \neq j$.

To begin with, we used uniform crossover [17] in the GA. A difficulty arises in using uniform crossover when cost optimal solutions are desired for a specified number of supports, n_{req} . Even if the number of supports in the parents is equal to n_{req} , the offspring generated using uniform crossover do not necessarily have exactly n_{req} supports. This necessitates the imposition of a penalty on solutions having a number of supports not equal to n_{req} . It is undesirable to do so with respect to the performance of genetic algorithms. Therefore, we developed a new crossover scheme that avoids the use of a penalty function by implicitly accounting for the constraint on the number of supports. This crossover scheme requires that the seeding be done in accordance with Equation 3.5. This crossover, illustrated in Figure 3.3, is described as follows. If the two parent strings that have exactly

n_{req} supports are given by $\langle b_{11}, b_{12}, \dots, b_{1n} \rangle$ and $\langle b_{21}, b_{22}, \dots, b_{2n} \rangle$, then consider the set of nodes given by $S = \{i : b_{1i} \neq b_{2i}\}$. Note that S , whose cardinality is always even, can be randomly divided into two sets of equal sizes S_1 and S_2 . Then, the offspring are given by:

$$o_{1i} = o_{2i} = b_{1i}, \forall i \notin S \quad (3.6)$$

$$o_{1i} = 1, \forall i \in S_1 \quad (3.7)$$

$$o_{2i} = 1, \forall i \in S_2 \quad (3.8)$$

The above scheme ensures that the number of supports in each offspring is equal to n_{req} .

This crossover scheme reduces the size of the search space to $nC_{n_{req}}$ from 2^n .

For mutation with uniform crossover, a randomly selected gene in the string can be modified according to the string representation used. In the integer representation, its value is modified randomly such that Equation 3.2 is satisfied. In the binary representation, its value is taken as the complement of the current value. For mutation with the new crossover technique proposed above, the mutation operator is required to ensure that the resulting solution has exactly n_{req} supports. Therefore, a mutation technique is chosen that removes a support from one of the nodes that originally had a support, and adds one to a randomly chosen node that previously did not have a support. The objective function and the selection technique used are described in the following section.

3.4 Cost optimization

Minimization for number of supports as well as the cumulative support capacity [7] in a single objective appears inappropriate primarily because the two quantities represent competing criteria. For example, minimization for number of supports yields few high capacity supports whereas minimization for cumulative support capacity yields a larger number of low capacity supports. A designer, under such circumstances, could benefit from a trade-off curve between the two. Further, an explicit understanding of the effect of capital and lifetime costs cannot be gained from the presently used model.

The effects of cost, both capital and lifetime cost, are examined with trade-off studies between the corresponding cost and the number of supports. We study the trade-off between the cost and the number of supports by considering an objective function that is similar in purpose to the one used by Chiba et al. [7]. The only exception being that the cost is included directly instead of minimization for the number of supports and cumulative support capacity. For the purpose of studying trade-off, the minimum cost solutions for a specified number of supports n_{req} are evaluated by using a penalty when the number of supports in a particular solution are other than the specified. It should be noted that this penalty becomes redundant when the seeding is done according to Equation 3.5 and the crossover scheme used is as defined by Equations 3.6, 3.7 and 3.8. The objective function is defined as

$$\max \left\{ \frac{1}{a\Delta\delta + b\Delta\sigma + c\Delta\theta + dC + e|n_{sup} - n_{req}|} \right\} \quad (3.9)$$

$$\Delta\delta = \sum_{\substack{i=1 \\ s_{i \max} > s_{all}}}^{n_s} (s_{i \max} - s_{all}); \Delta\sigma = \sum_{\substack{j=1 \\ \sigma_j > \sigma_{all}}}^n (\sigma_j - \sigma_{all}); \Delta\theta = \sum_{\substack{k=1 \\ \theta_k > \theta_{kc}}}^{n_{sup}} (\theta_k - \theta_{kc}) \quad (3.10)$$

where a, b, c, d, e are multipliers that are assigned values according to the relative importance of the constraints; σ_i is the stress at node i ; σ_{all} is the allowable stress; n is the total number of nodes in the piping system; n_s is the total number of spans in the system; $s_{j \max}$ is the maximum displacement to span ratio in span j ; s_{all} is the allowable displacement to span ratio; θ_k and θ_{kc} are the support load and support capacity, respectively, at the k^{th} support location; n_{sup} is the total number of supports in the system; and n_{req} is the number of supports specified for a given solution. The following subsection describes the piping system chosen to illustrate cost (capital and lifetime cost) optimization using the proposed GA.

3.4.1 Example piping system

For simplicity, a 46 cm (18 inch) outer diameter straight pipe having 3.5 cm (1.375 inch) wall thickness is considered. As shown in Figure 3.1, the two ends of the straight pipe are anchored and several lumped masses representing valves and other equipment are located on the pipe. In Figure 3.1, the numbers under the lumped masses show their corresponding weights in kips. The pipe is discretized into beam elements with 46 nodes including the two anchors. Each node on the pipe is allowed only two degrees of freedom, namely, the vertical displacement and the in-plane rotation. The piping system is modeled on commercial piping analysis program PIPESTRESS [8]. Seismic analysis is performed for a

response spectrum input in the vertical direction corresponding to El Centro, 1940 S00E earthquake that is applied uniformly at all the supports. Figure 3.2 shows the flowchart illustrating the interface of PIPESTRESS with Vitri for GA-based optimization. Table 3.2 shows the list of supports that are considered for the piping system, along with their cost and support capacities. The following subsection describes the application of the proposed GA for capital cost optimization by using this piping system as an example.

3.4.2 Capital cost optimization

The capital cost for various support types can be directly considered in the formulation because such costs are readily available. The capital cost is given by

$$C = \sum_{i=1}^{n_{sup}} c_i \quad (3.11)$$

where c_i is the capital cost of support at location i . The above equation is used to compute C in Equation 3.9. Tournament selection and stochastic universal selection [17] are tried and both appear to perform equally well for capital cost optimization.

It is necessary to verify that the solution generated by the genetic algorithm is the real optimal solution or very close to the real optimal solution in order to make sure that the proposed GA works for this problem. The real optimal solution for a given n_{req} can be found by evaluating all possible solutions having n_{req} supports, which actually count to $nC_{n_{req}}$. It must be noted that this is possible only for small n_{req} as the search space increases drastically with n_{req} . In this manner, the real optimal solutions for $n_{req} = 3, 4, 5$ are found.

The performance of the two crossover schemes is compared in Figure 3.4 with the benchmark as the real optimal solutions for $n_{req} = 3, 4, 5$. The points in the graph represent the best solutions that are generated over 5 runs of the genetic algorithm for each n_{req} . It is observed that the proposed crossover appears to produce better solutions, compared to the uniform crossover. Moreover, the proposed crossover scheme generates the real optimal solutions for $n_{req} = 3, 4, 5$. A notable difference between the two curves is that unlike the proposed crossover, uniform crossover could not find the optimal solution for $n_{req} = 3$ as only a very few of the solutions with three supports are feasible.

The optimal solutions generated using proposed crossover, the cost values of which are plotted in Figure 3.4, and their important properties for various n_{req} are shown in Figure 3.6 and Table 3.3 respectively. The (θ_k/θ_{kc}) values of these solutions are shown in Figure 3.9. Figure 3.4 also shows that the minimum cost solutions have minimum number of supports. It should be noted that in an integer string representation, minimization for only the number of supports without explicitly considering the cost in the objective function would not yield the same solution. Several different feasible solutions, with different support types but equal number of supports would exist, each having a different cost. On the contrary, minimization for the number of supports in a binary string representation will produce a least cost solution as the sizing for support type can be delayed until the final step in each evaluation. Next, we study the effect of lifetime cost rather than just the capital cost.

3.4.3 Lifetime cost optimization

Lifetime cost includes the cost of replacement and outage in addition to the capital and maintenance cost. Revenue losses due to unscheduled outage are much higher than the maintenance and capital costs. Since specific estimates of lifetime costs are unavailable, lifetime cost is represented as being proportional to some power of the support capacities, i.e.,

$$C = \sum_{i=1}^n c_i^f \quad (3.12)$$

We consider $f = 1.2$ even though actual values of f are expected to be much higher. Stochastic universal selection is used in the GA as it appeared to perform better than tournament selection for this fitness function.

The results generated by using this cost function in the fitness function given by Equation 3.9 are shown in Figure 3.5. The figure shows two curves, one representing the results generated using uniform crossover and the other using the proposed crossover. The solutions represented by both the curves are the best solutions generated over 5 different runs of the GA for each n_{req} . The figure also shows the lifetime costs of the real optimal solutions generated by enumeration, as described earlier for capital cost, for $n_{req} = 3, 4, 5$. The proposed crossover scheme produces significantly better results due to reduced search space and as in the case of capital cost, the proposed crossover scheme matches the real optimal solutions for $n_{req} = 3, 4, 5$. In contrast to the results shown in Figure 3.4, it can also be observed that there is a trade-off between the lifetime cost and the number of supports even though we considered a relatively small value of f .

The optimal solutions generated by proposed crossover are shown in Figures 3.7 and 3.8. The important properties of these solutions are given in Tables 3.5 and 3.5. The (θ_k/θ_{kc}) values of the solutions in Figure 3.7 are shown in Figure 3.10. It is observed from the optimal solutions for both, capital and lifetime cost, that the desire of having the supports under lumped masses is not satisfied. In the following section, a lumped mass parameter U is proposed to model this desirability as an objective in the GA.

3.5 Modified objective function

The optimal solutions for capital cost and lifetime cost corresponding to different values of n_{req} , are shown in Figures 3.6, and 3.7 and 3.8 respectively. It can be noted that most of the supports in these solutions are not under the lumped masses. This is so because the requirements for lumped masses has not been considered either explicitly or implicitly in the objective function.

Consideration of support locations to be under the lumped masses requires some means of quantification. While placing a support under all the heavy lumped masses would be cost intensive, it is desirable to have as many lumped masses supported as possible. To do so, we propose a lumped mass parameter U as follows:

$$U = \left(\frac{m_u}{m} \right) \left(\frac{n_u}{n_{sup}} + 1 \right) \quad (3.13)$$

where m_u is the total unsupported lumped mass and m is the total lumped mass in the solution. n_u is the number of supports not under lumped masses and n_{sup} is the total

number of supports in a given solution.

The objective function given by Equation 3.9 can be modified to include U as follows:

$$\max \left\{ \frac{1}{a\Delta\delta + b\Delta\sigma + c\Delta\theta + dC + e|n_{sup} - n_{req}| + fU} \right\} \quad (3.14)$$

where f is a constant chosen appropriately to obtain solutions in which the heavier masses in the piping system are supported. Figure 3.11 shows the solution obtained by using Equation 3.14 as the objective function. The cost of such solutions, given in Table 3.7, are much higher than the cost of the optimal solutions for the corresponding n_{req} in Figure 3.5. It is desirable to consider only small deviations from the optimal cost (about 10% to 20%) for accommodating issues such as the location of supports under lumped masses. Consequently, a designer would benefit by generating alternatives that not only have costs in the neighborhood of the cost optimal solution but also low values of U . Next, we explore the role of a technique referred to as “Modeling to Generate Alternatives (MGA)” for evaluating alternative solutions.

3.6 Modeling to Generate Alternatives (MGA)

A traditional use of optimization is to create and run a model to find the “optimal” solution. This solution, however, is rarely the best solution to the real problem because some objectives and constraints may not have been explicitly stated in the problem formulation. The omissions may be due to errors, the unquantifiable nature of certain issue or issues that are not identified at the point of model formulation. Traditionally, this has been a major

limitation of optimization applications. Modeling to Generate Alternative (MGA) techniques [1, 4] use optimization to generate a small set of very different solutions. MGA is 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.” By generating these good yet different solutions, a decision maker can explore alternatives that satisfy unmodeled objectives to varying degrees. These alternatives may be assessed by the decision maker either subjectively or quantitatively.

For instance, consider the problem of optimizing a single objective Z_1 . Let it have a solution A as shown in Figure 3.12A. The addition of another objective Z_2 would result in a trade-off, probably like the one shown in Figure 3.12B. Solution B , which is inferior to A on the omission of Z_2 , might be the most desirable compromise between Z_1 and Z_2 for the decision maker. In a similar manner, there might be a solution C which is not on the trade-off curve but is actually preferable to the designer in the presence of a third objective Z_3 .

To further illustrate this idea, consider the following formulation of a design optimization problem:

Maximize $z = x + 2y$ subject to

$$x + y \leq 30$$

$$y \leq 20$$

$$x, y \geq 0$$

The decision space for the problem is shown as the shaded region in Figure 3.13A 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 $\langle x, y \rangle = \langle 10, 20 \rangle$. The premise of MGA, however, is that all other things may not be equal, and that there may be problem 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 3.13B, 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 inspection.

Baugh et. al. [1] illustrate these ideas with a topological truss optimization. Other MGA techniques have been extensively used in environmental engineering applications for air

pollution control as well as for land use and planning [6, 5, 19]. For the problem given earlier, the first alternative can be found by changing the formulation to maximize a difference metric δ , instead of z and imposing the following constraint:

$$x + 2y \geq 0.9(50)$$

This constraint ensures that the alternative generated is a “good” solution by searching the objective space around the optimal solution. δ for this problem can be given by:

$$(x - 10)^2 + (y - 20)^2$$

which is the distance between the two points in the decision space. In this manner, an alternative that is very different from the optimal solution in decision space but yet closer to the optimal in objective space can be obtained.

3.7 Application of MGA to pipe support optimization

As noted from the discussion in the previous section, application of MGA requires definitions of a reduced decision space and a difference metric to generate the alternatives. For pipe support optimization, the decision space can be reduced by imposing the following constraints:

$$C_{mga} \leq k_1 C_{opt} \quad (3.15)$$

$$fitness_{mga} \geq k_2 fitness_{opt} \quad (3.16)$$

$$U_{mga} \leq U_{all} \quad (3.17)$$

where $k_1 > 1$ and $k_2 < 1$; C_{mga} and C_{opt} are the cost of the alternative and the cost of the optimal solution respectively; $fitness_{mga}$ and $fitness_{opt}$ are the fitness values of the alternative and the optimal solution respectively; U_{all} is an upper bound on the lumped mass parameter specified by the designer, and U_{mga} is the value of the lumped mass parameter for an alternative solution; r_{max} is not considered for the MGA as it is found to be very restrictive for the generation of alternatives as indicated in section 4.4.

Topology metric is used as the difference metric in the present study. This metric, commonly known as ‘‘Hamming distance,’’ is applicable for binary strings. If the two strings are given by $\langle b_1, b_2, \dots, b_n \rangle$ and $\langle c_1, c_2, \dots, c_n \rangle$, the difference metric is given by:

$$\delta = \sum_{i=1}^n |b_i - c_i| \quad (3.18)$$

The flowchart showing the incorporation of MGA in Vitri is given in Figure 3.14. The alternatives generated using the topology metric in the MGA for $n_{req} = 5$ are given in Figure 3.15. The important characteristics of the alternatives are listed in Table 3.8. The values that are used for the MGA parameters in Equations 3.15, 3.16, 3.17 to produce these results are as follows: $U_{all} = 1.8$, $k_1 = 1.25$ and $k_2 = 0.8$. More lumped masses are supported in the alternatives than in the optimal solutions given in 3.7. It is seen that all the alternatives are very different from each other in topology. These solutions, which are spread over a large expanse of the decision space, provide the decision maker with an opportunity for tinkering with the solutions to generate better solutions. The alternatives generated for $n_{req} = 7$ are shown in Figure 3.16. The characteristics of these solutions are

given in Table 3.10. A stricter constraint on U_{all} is possible for $n_{req} = 7$ as there are more number of supports. The alternatives for $n_{req} = 7$ are generated using $U_{all} = 1.000$, $k_1 = 1.3$ and $k_2 = 0.8$. It should be noted that $C_{opt} = 1269610$ because the optimal solution with respect to lifetime cost is that of $n_{req} = 5$ as seen from Figure 3.5.

In the process of generating these alternatives, various values for U_{all} and k_1 are tried. It is seen that the chances of finding alternatives decrease rapidly as the values of U_{all} and k_1 are reduced. This observation supports the conclusion reached upon the end of the examination of the modified fitness function that it is excessively costly when all the heavier lumped masses are supported. At the same time it should be kept in mind that this piping system had a large number of heavy masses for the purpose of studying the effects of these parameters. Actual piping systems may also have a large number of masses but they are typically comparable in size to each other. Thus, there is a better chance of generating much better solutions when the same is applied to a real piping system.

3.8 Application to real life piping system

The developed techniques are applied to the real life piping system shown in Figure 3.17. This piping system is in use in a nuclear power plant operated by Carolina Power & Light. It consists of a 12.75 inch diameter Schedule 100 pipe loop that is anchored at the reactor vessel. This loop is connected to an 18 inch diameter Schedule 120 pipe that is anchored at the bottom of the concrete shield wall. Like in the earlier piping system,

El Centro is used as the earthquake input to the piping system. The piping system has 75 support locations and each location can have a maximum of three supports, namely, one in each of the principal directions x , y and z . Some new issues that arise during the analysis of the real life piping system are discussed in section 4.5.

The results of the trade-off study between lifetime cost and the number of supports for this piping system is given in Figure 3.18. Even though the trough of the trade-off curve seems to be near $n_{req} = 25$, the fundamental frequency of the piping system is much higher than 5 Hz, because of the higher stiffness of the system. Designers normally prefer the frequency of the piping system to be close to 5 Hz. The number of supports required to have such a frequency is more likely to be near $n_{req} = 10$.

3.9 Summary and conclusions

A GA-based approach for seismic design and performance evaluation of pipe supports is discussed in the paper. Supports are modeled as rigid springs, permitting the use of a binary string representation in the GA. The change in representation from integer to binary string is shown to produce a considerable reduction in the size of the search space to be explored by the GA. Alternative seeding techniques to create populations, in which the number of supports in its solutions is nearer to the required number, are explored. The trade-off curves between various types of cost (capital or lifetime) and the number of supports are obtained. A new crossover scheme is proposed to aid in the generation of

optimal solutions having a specified number of supports. The performances of the proposed crossover scheme and uniform crossover are compared. A proposed lumped mass parameter, U , is used in the MGA technique to generate useful alternatives. Finally, the practical applicability of the developed techniques is shown by applying them to a real piping system.

The study shows that the solutions are primarily infeasible with respect to support capacities, and not because of the stress and the displacement constraints. Binary string representation performs better than the integer string representation but it should be noted that it is applicable only if the supports are modeled as rigid springs. The results show that the proposed crossover scheme works better than uniform crossover. Optimization for capital cost is same as minimizing the number of supports whereas optimization for lifetime cost is a multi-objective problem. The results from the use of the modified fitness function show that it might be practically infeasible to support all the heavy lumped masses of a pipe because of the high cost involved. MGA, using topology metric for difference and the proposed lumped mass parameter as a constraint to reduce the decision space appears to work well.

Table 3.1: Location of lumped masses along the pipe

SI No.	Distance from left end of the pipe (in ft.)
1	12
2	18
3	30
4	42
5	54
6	63
7	75
8	87
9	102
10	111
11	129

Table 3.2: Cost and capacity of supports

Sl No	Support capacity(lbs)	Cost(dollars)
0	0	0
1	1978	15000
2	2067	15100
3	3600	21000
4	8610	21500
5	13750	24000
6	19150	24600
7	20100	24800
8	23450	32000
8	37300	35500
9	62000	36000
10	70350	42000

Table 3.3: Properties of optimal capital cost solutions given in Figure 3.6

n_{req}	Cost (in 10,000s)	U	$\left(\frac{\theta_k}{\theta_{kc}}\right)_{max}$	Frequency (Hz)	$\left(\frac{\sigma_i}{\sigma_{all}}\right)_{max}$
3	108.000	0.873	0.921	5.489	0.980
4	131.600	1.448	0.951	9.632	0.593
5	162.900	2.000	0.952	9.398	0.429
6	189.000	0.930	0.974	8.768	0.457
7	203.300	1.122	0.938	8.760	0.457
8	218.300	1.146	0.977	8.753	0.459
9	232.700	0.803	0.991	9.580	0.492

Table 3.4: Anchor forces of optimal capital cost solutions given in Figure 3.6

n_{req}	Anchor 1		Anchor 2	
	F_y (in lbs)	M_z (in lb-ft)	F_y (in lbs)	M_z (in lb-ft)
3	68507	632607	56389	383012
4	20010	120032	57074	382689
5	10150	46519	44139	276992
6	8328	32585	50560	295129
7	412	1409	50614	295151
8	1136	4	50739	296099
9	424	1275	47203	317243

Table 3.5: Properties of optimal lifetime cost solutions given in Figures 3.7 and 3.8

n_{req}	Cost (in 10,000s)	U	$\left(\frac{\theta_k}{\theta_{kc}}\right)_{max}$	Frequency (Hz)	$\left(\frac{\sigma_i}{\sigma_{all}}\right)_{max}$
3	169.0406	0.873	0.921	5.489	0.980
4	131.3528	1.448	0.951	9.632	0.593
5	126.9610	2.000	0.952	11.007	0.244
6	131.4293	1.365	0.995	13.055	0.096
7	135.4763	2.000	0.988	11.720	0.257
8	143.7118	1.289	0.987	10.985	0.098
9	152.3744	1.840	0.995	15.860	0.157

Table 3.6: Anchor forces of optimal lifetime cost solutions given in Figures 3.7 and 3.8

n_{req}	Anchor 1		Anchor 2	
	F_y (in lbs)	M_z (in lb-ft)	F_y (in lbs)	M_z (in lb-ft)
3	68507	632607	56389	383012
4	20010	120032	57074	382689
5	22687	136491	31272	157784
6	10298	46949	13769	53662
7	27542	166163	13777	54162
8	10724	49855	12969	59132
9	7599	29075	21149	101047

Table 3.7: Properties of optimal solutions obtained using Equation 3.14

n_{req}	Cost (in ten thousands)	U
6	244.8697	0.102
7	253.3298	0.077
8	202.6044	0.060

Table 3.8: Properties of MGA solutions for $n_{req} = 5$ in Figure 3.15

Sl. No.	Cost (in 10,000s)	U	$\left(\frac{\theta_k}{\theta_{kc}}\right)_{max}$	Frequency (Hz)	$\left(\frac{\sigma_i}{\sigma_{all}}\right)_{max}$
1	149.7216	1.099	0.991	9.953	0.547
2	153.1160	1.785	0.957	9.493	0.365
3	140.0385	1.532	0.997	10.254	0.353

Table 3.9: Anchor forces of MGA solutions for $n_{req} = 5$ in Figure 3.15

Sl. No.	Anchor 1		Anchor 2	
	F_y (in lbs)	M_z (in lb-ft)	F_y (in lbs)	M_z (in lb-ft)
1	19646	108406	53240	352690
2	14238	67781	38970	235801
3	18894	103855	34972	227664

Table 3.10: Properties of MGA solutions for $n_{req} = 7$ in Figure 3.16

Sl. No.	Cost (in 10,000s)	U	$\left(\frac{\theta_k}{\theta_{kc}}\right)_{max}$	Frequency (Hz)	$\left(\frac{\sigma_i}{\sigma_{all}}\right)_{max}$
1	163.0442	0.576	0.964	18.324	0.084
2	154.4800	0.862	0.992	14.578	0.210

Table 3.11: Anchor forces of MGA solutions for $n_{req} = 7$ in Figure 3.16

Sl. No.	Anchor 1		Anchor 2	
	F_y (in lbs)	M_z (in lb-ft)	F_y (in lbs)	M_z (in lb-ft)
1	10974	50915	11610	38892
2	20952	114237	28109	135237

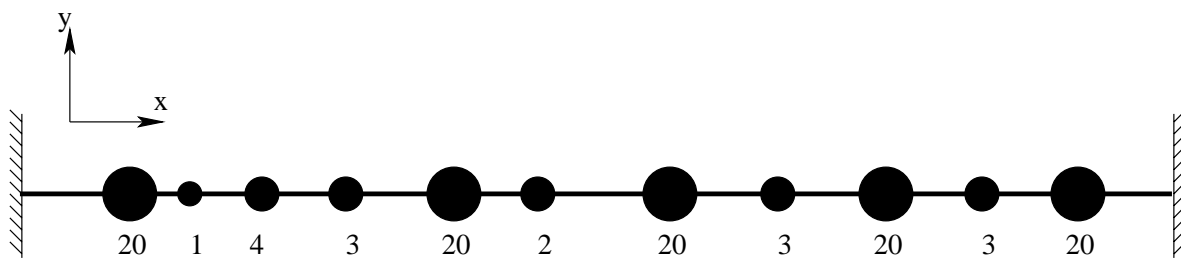


Figure 3.1: Straight pipe with lumped masses

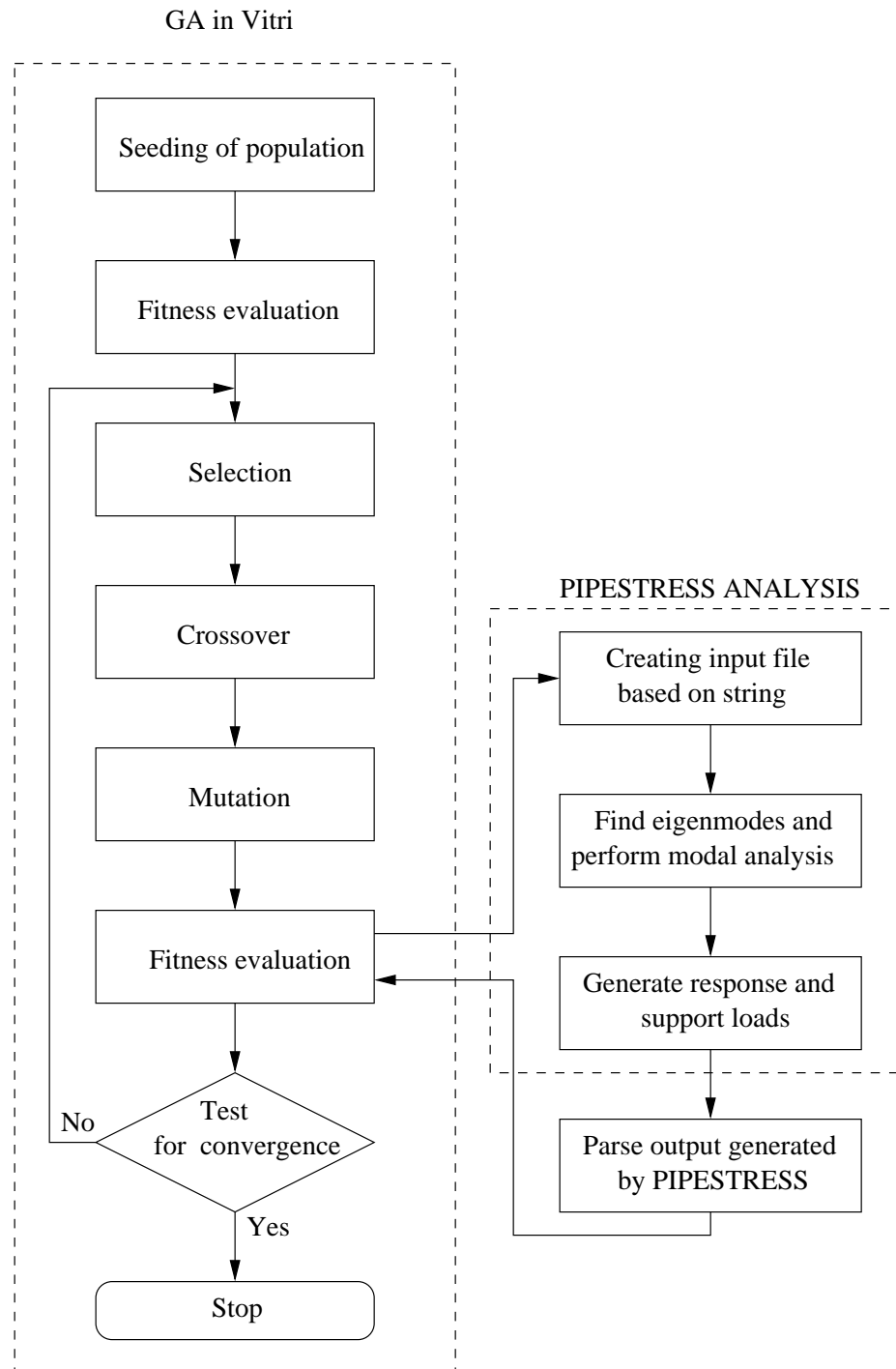


Figure 3.2: Interfacing of PIPESTRESS and Vitri

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

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

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

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

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

Figure 3.3: Illustration of the proposed crossover

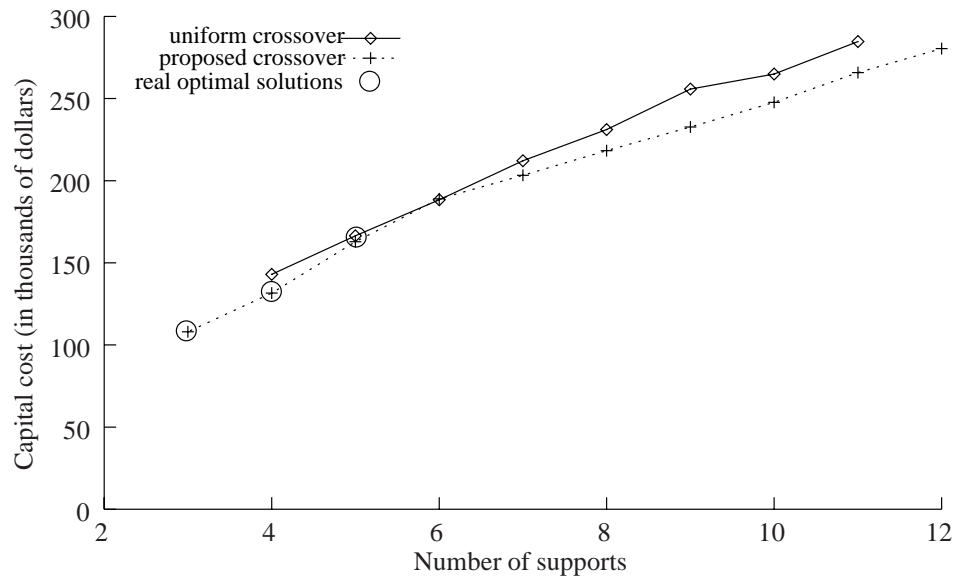


Figure 3.4: Results of multi-objective study for capital cost

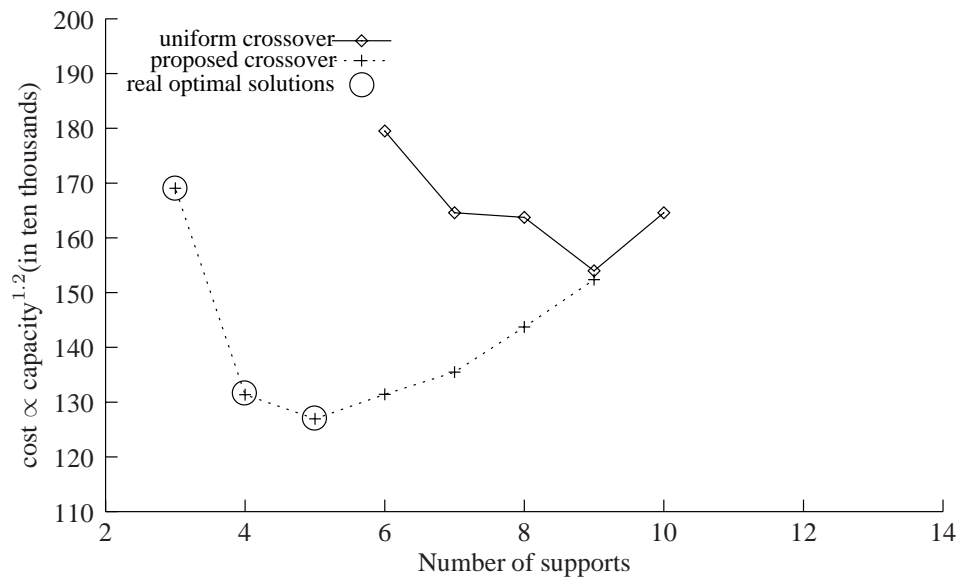


Figure 3.5: Results of multi-objective study for lifetime cost

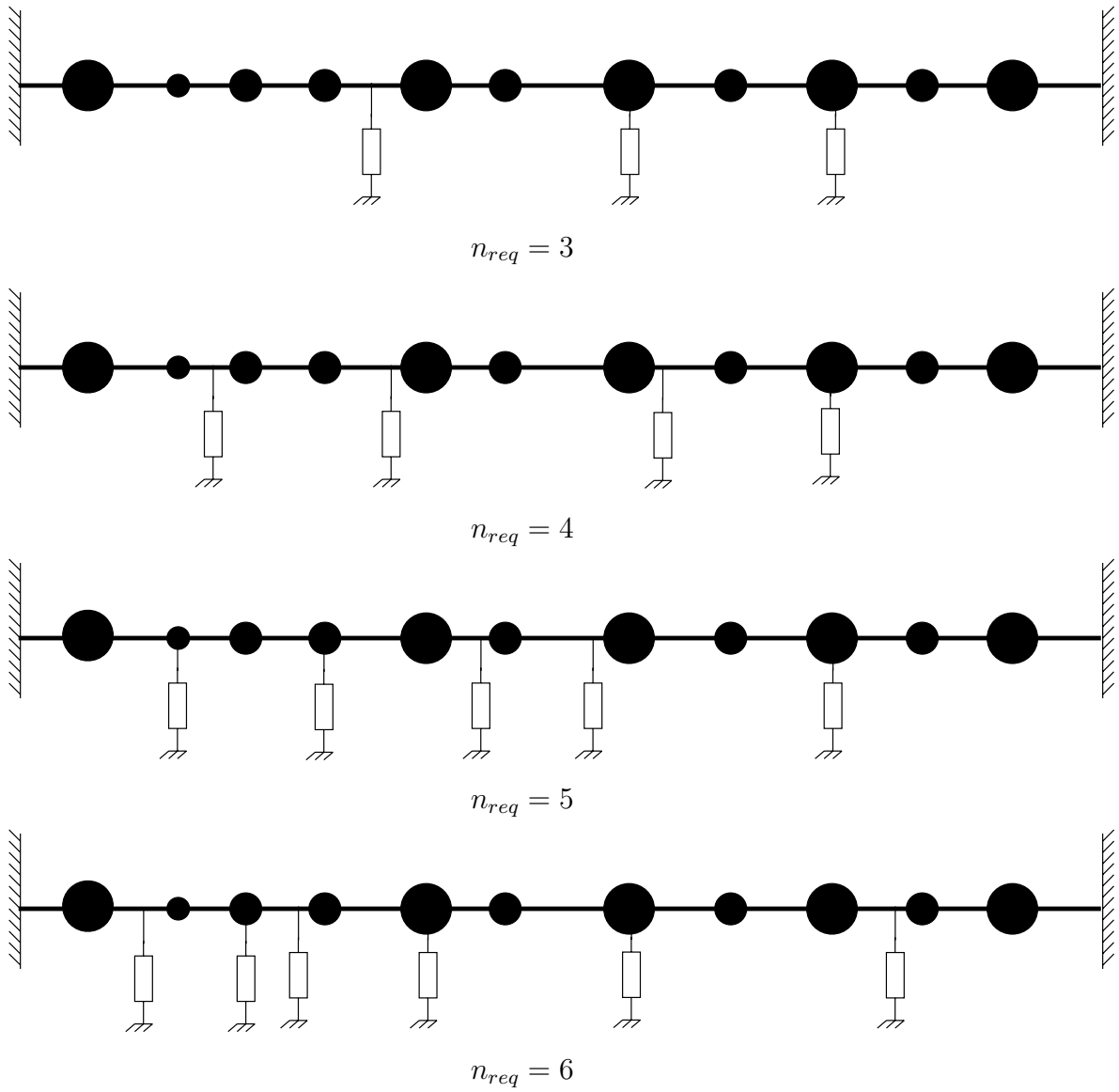


Figure 3.6: Optimal solutions for capital cost obtained using proposed crossover

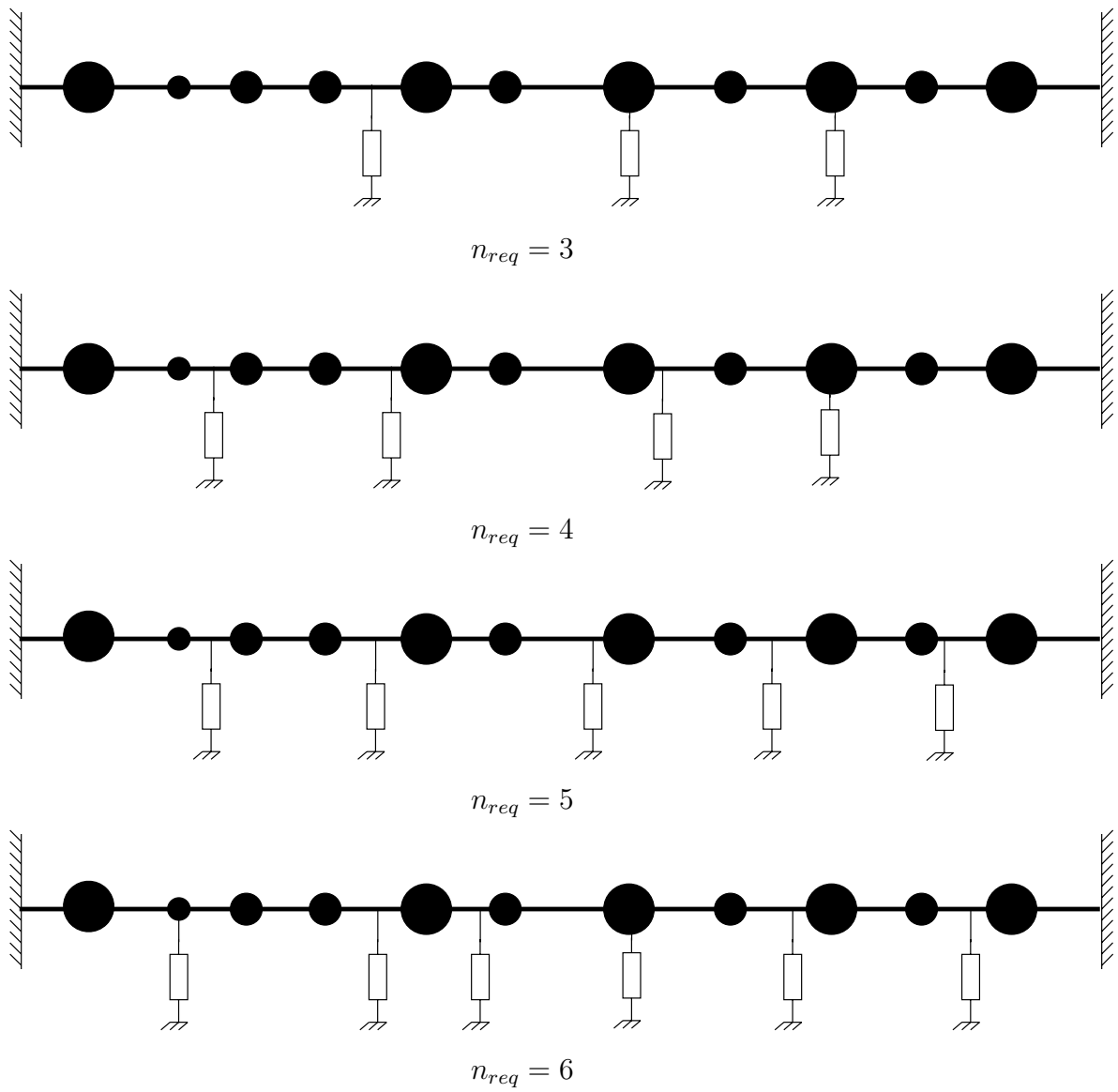


Figure 3.7: Optimal solutions for lifetime cost obtained using proposed crossover

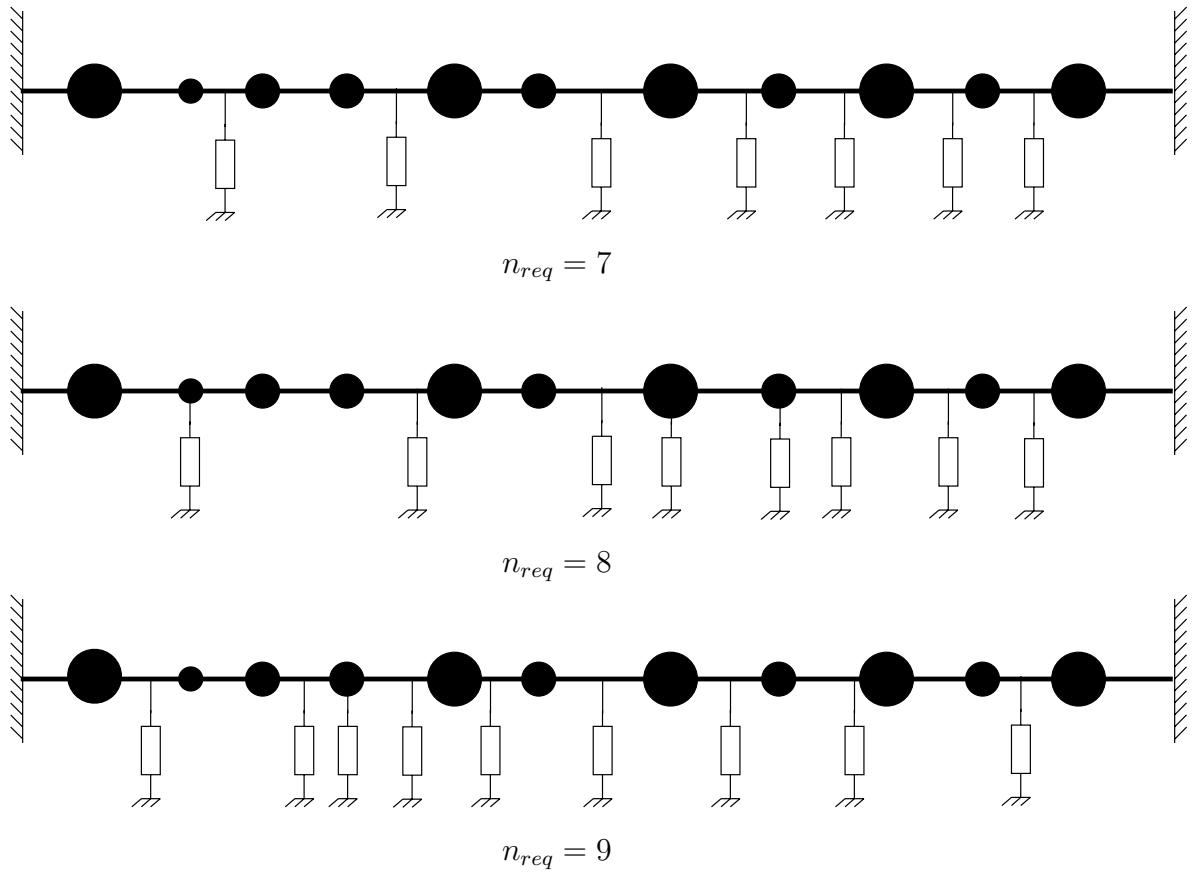


Figure 3.8: Optimal solutions for lifetime cost obtained using proposed crossover

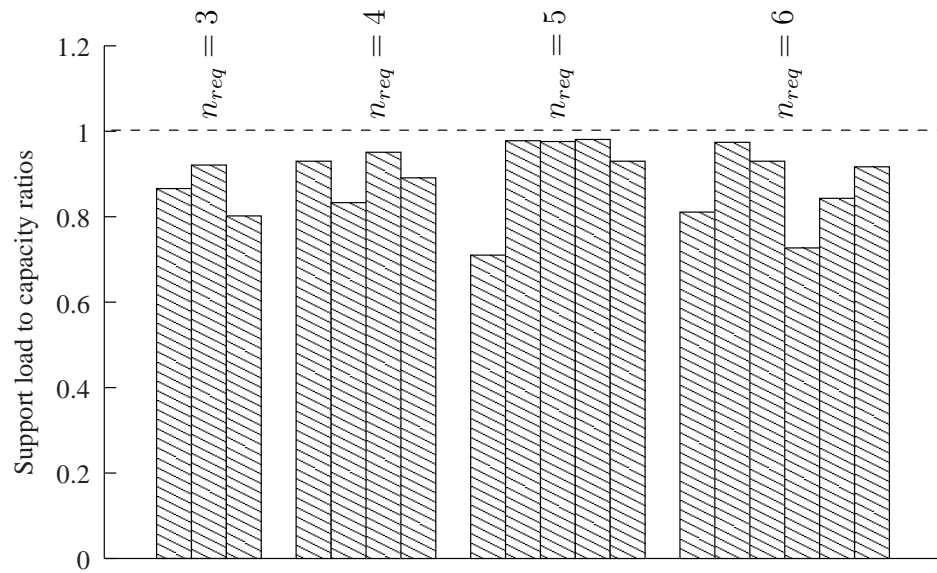


Figure 3.9: θ_k/θ_{kc} for supports of solutions in Figure 3.6

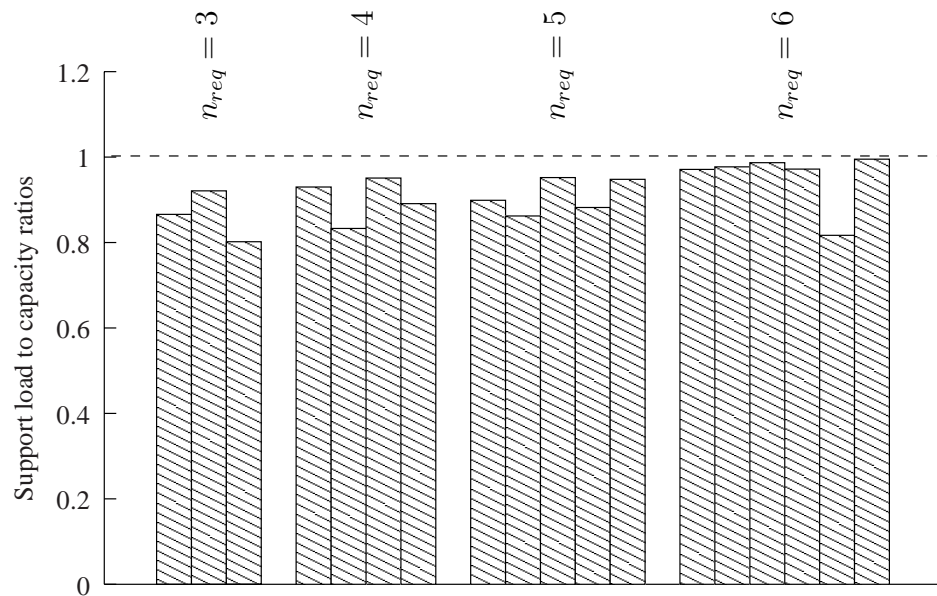


Figure 3.10: θ_k/θ_{kc} for supports of solutions in Figure 3.7

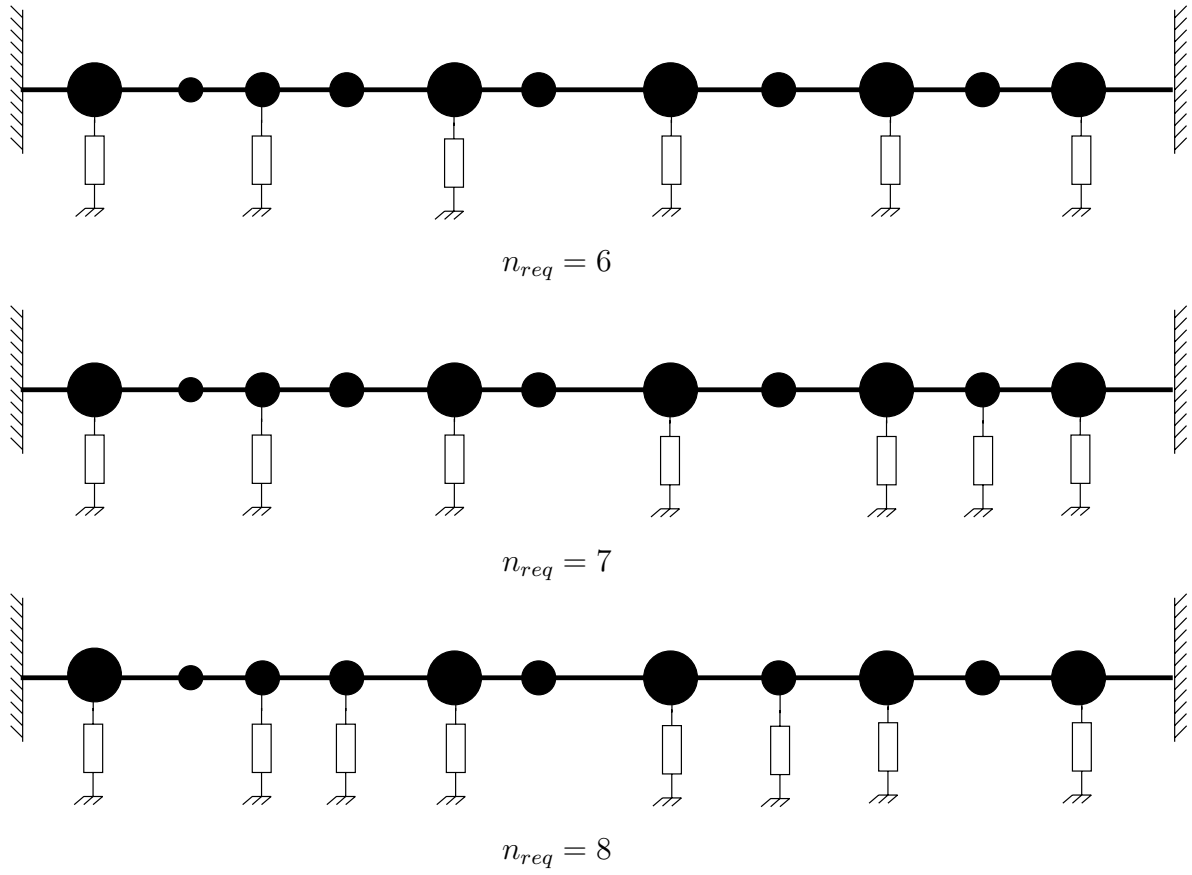


Figure 3.11: Best fitness solutions using modified fitness function (Equation 3.14) for life-time cost

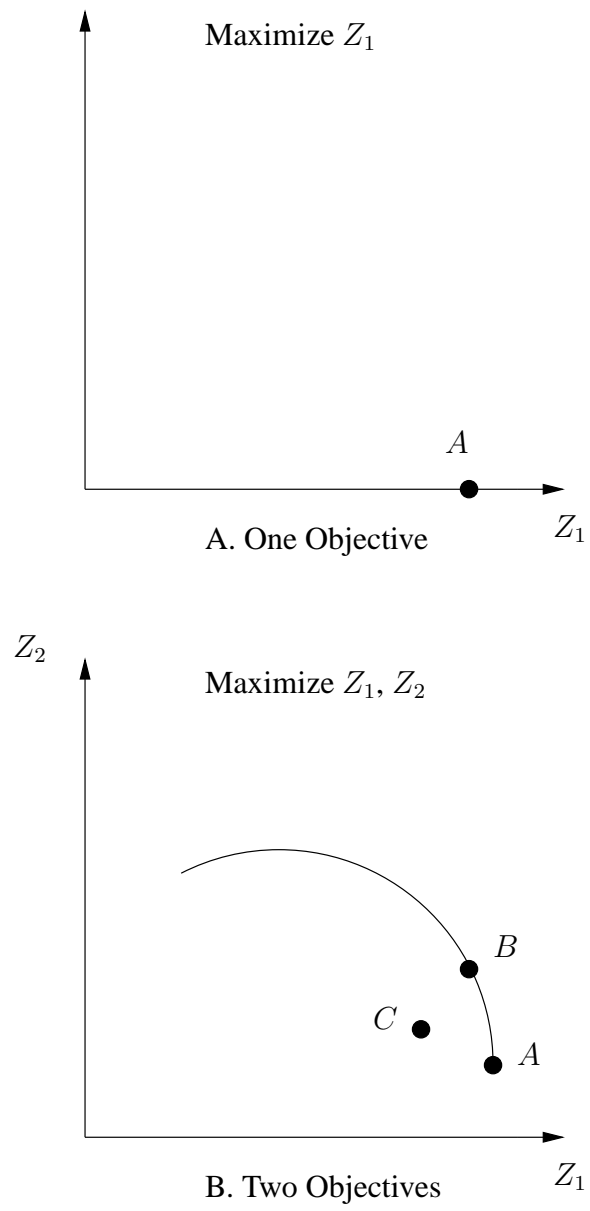


Figure 3.12: Multiple-objective optimization

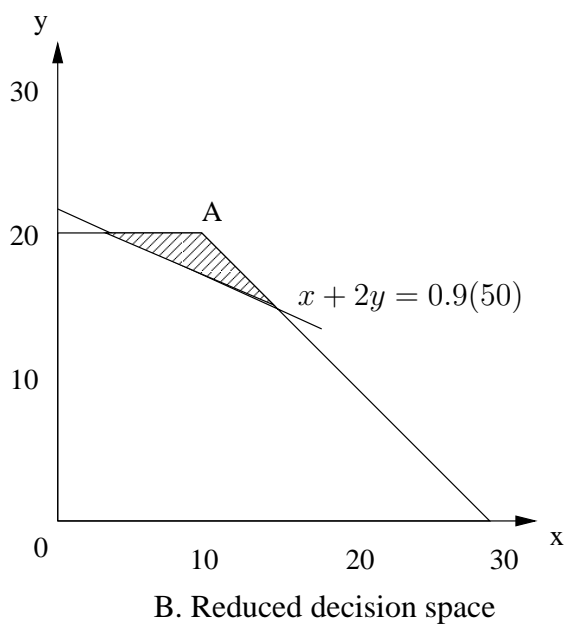
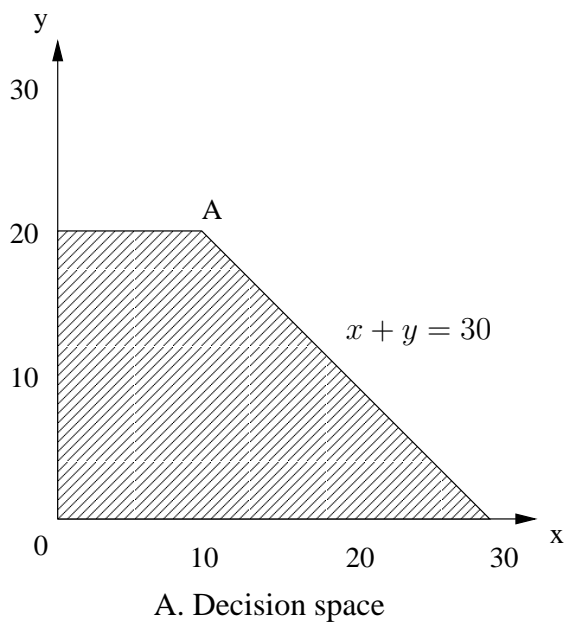


Figure 3.13: Decision space for LP example

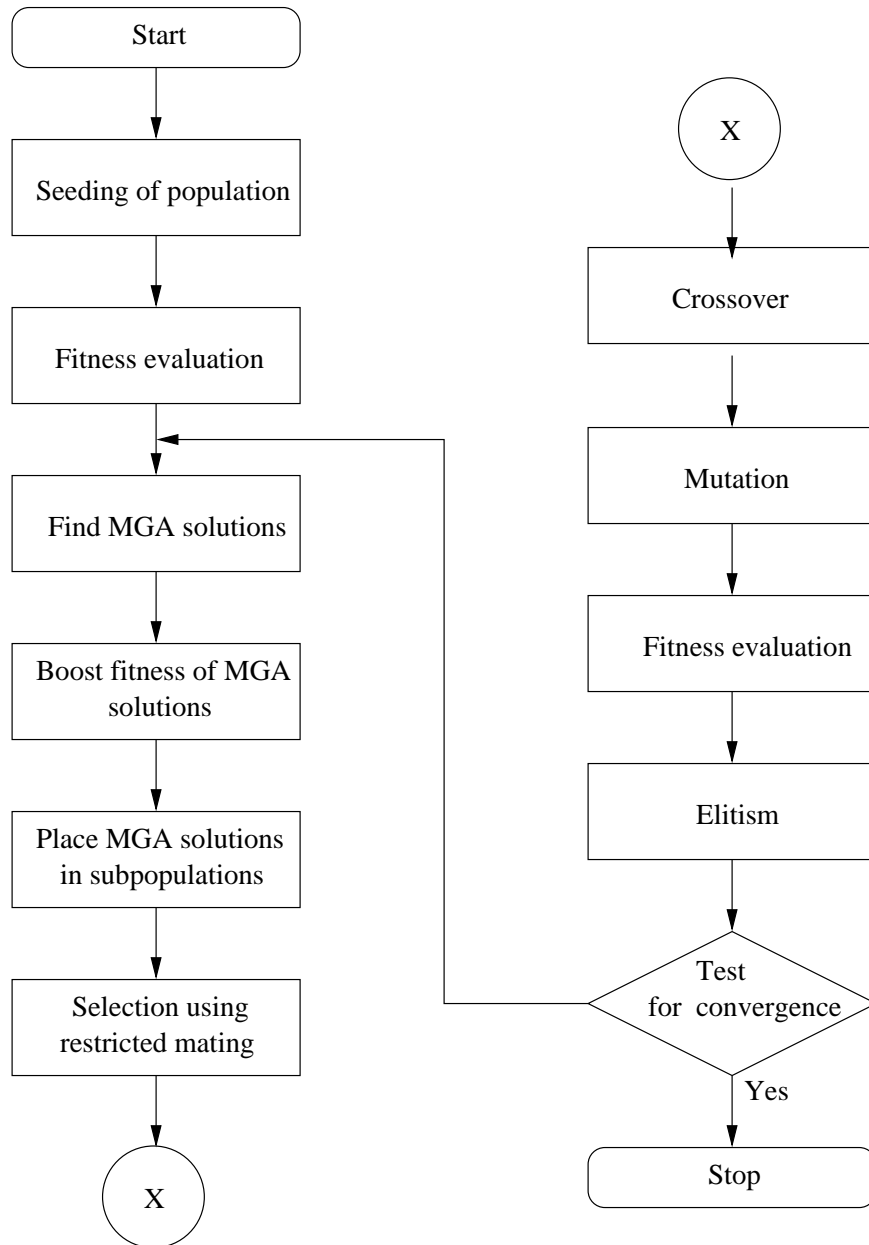


Figure 3.14: Flowchart showing incorporation of MGA in Vitri

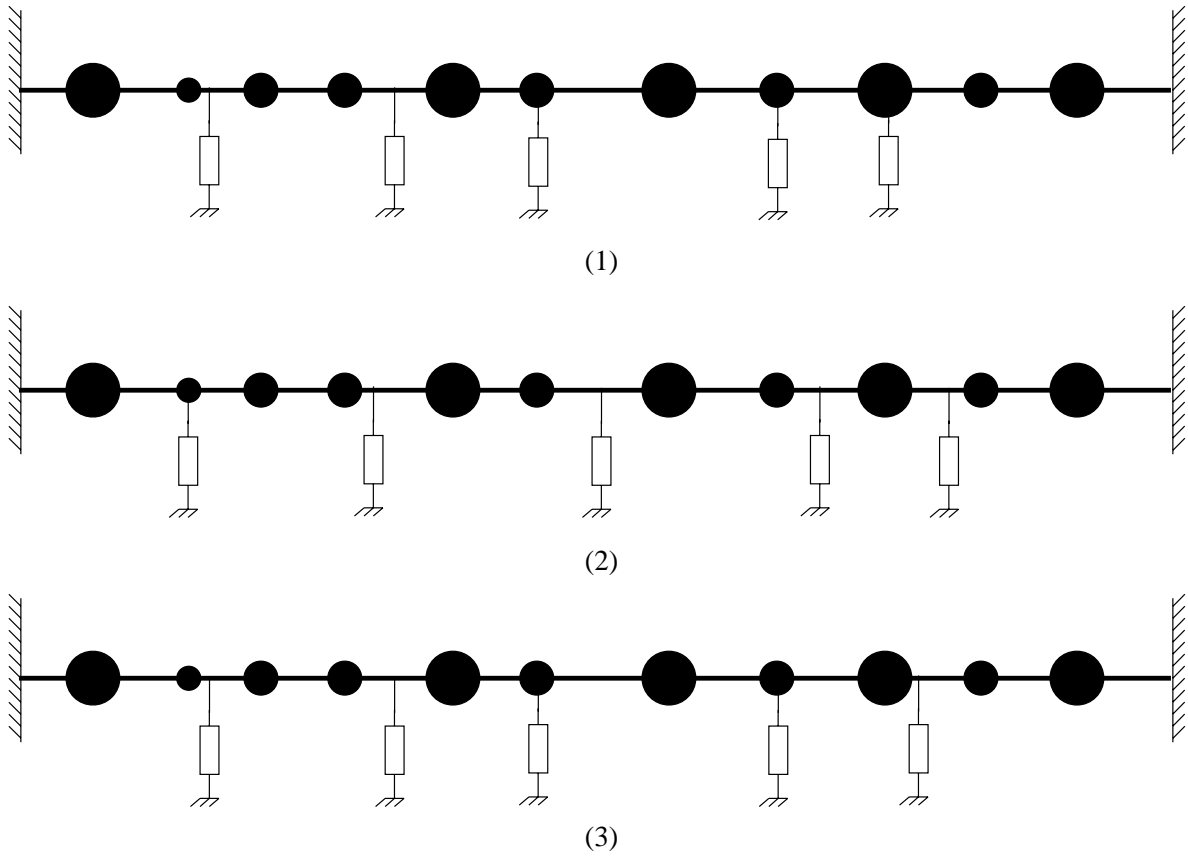


Figure 3.15: MGA solutions for lifetime cost with $n_{req} = 5$

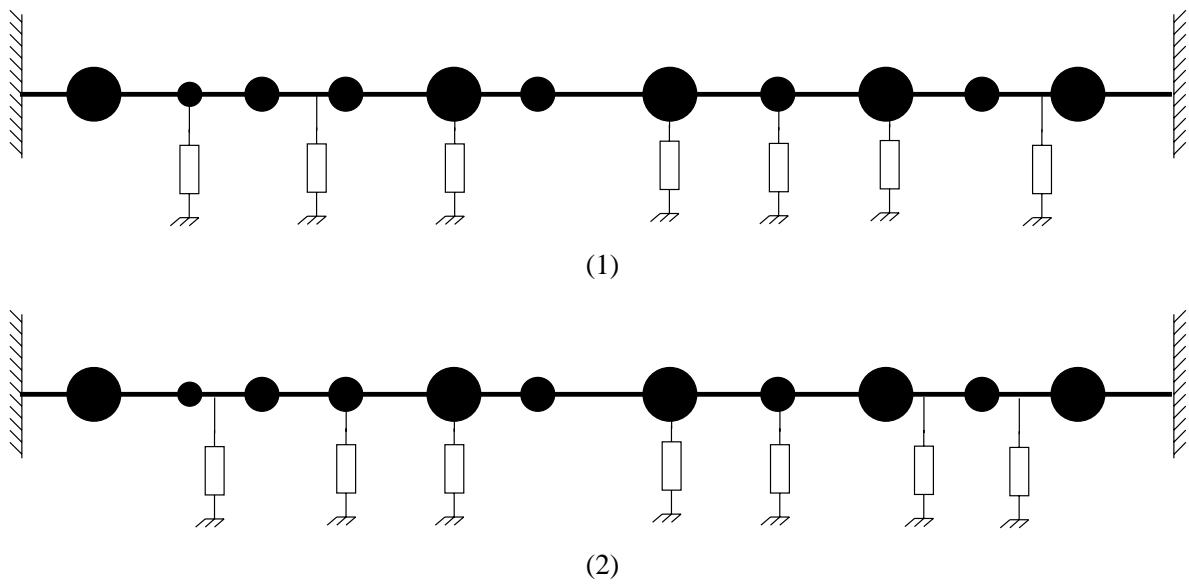


Figure 3.16: MGA solutions for lifetime cost with $n_{req} = 7$

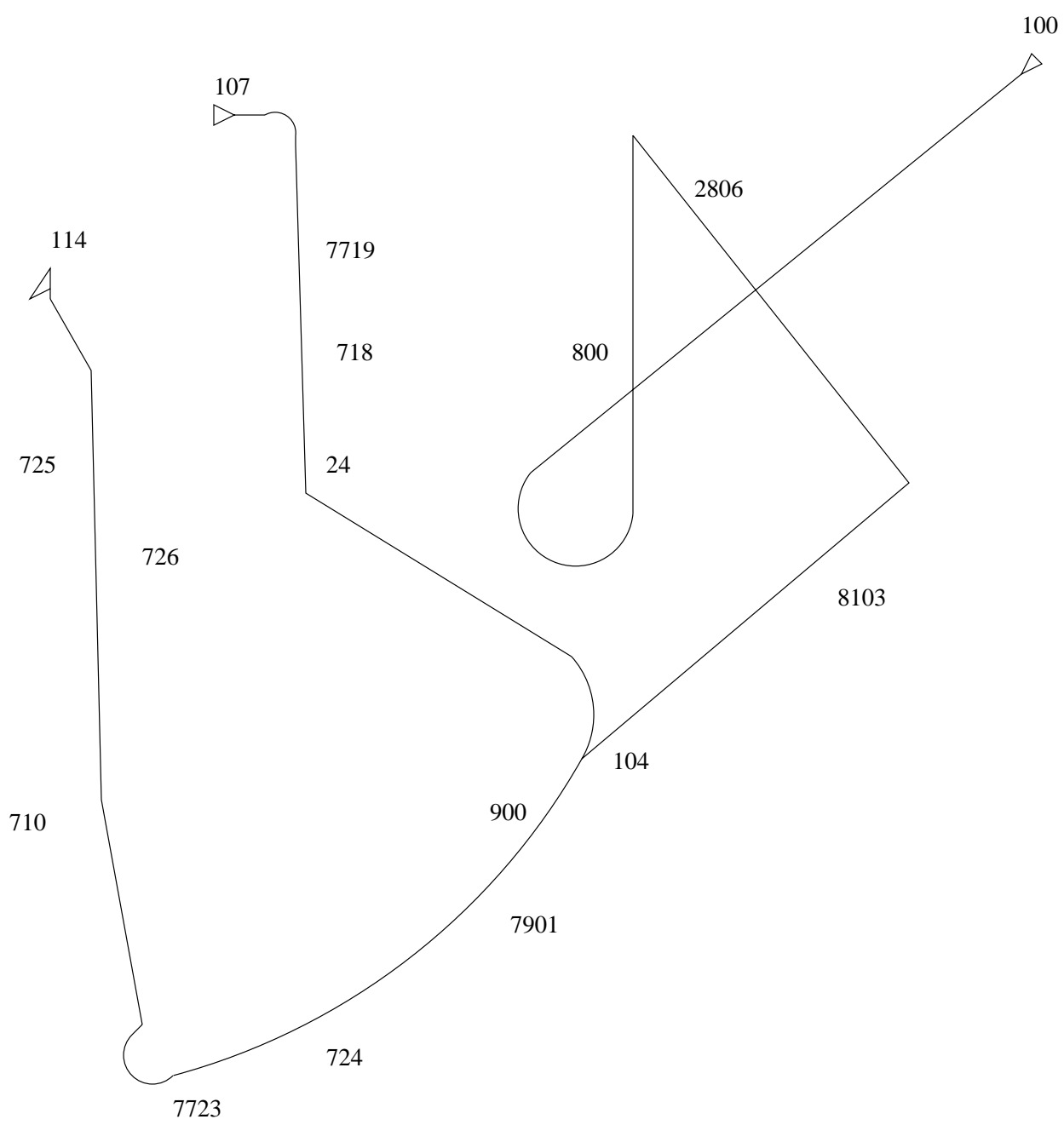


Figure 3.17: Real piping system (not to scale)

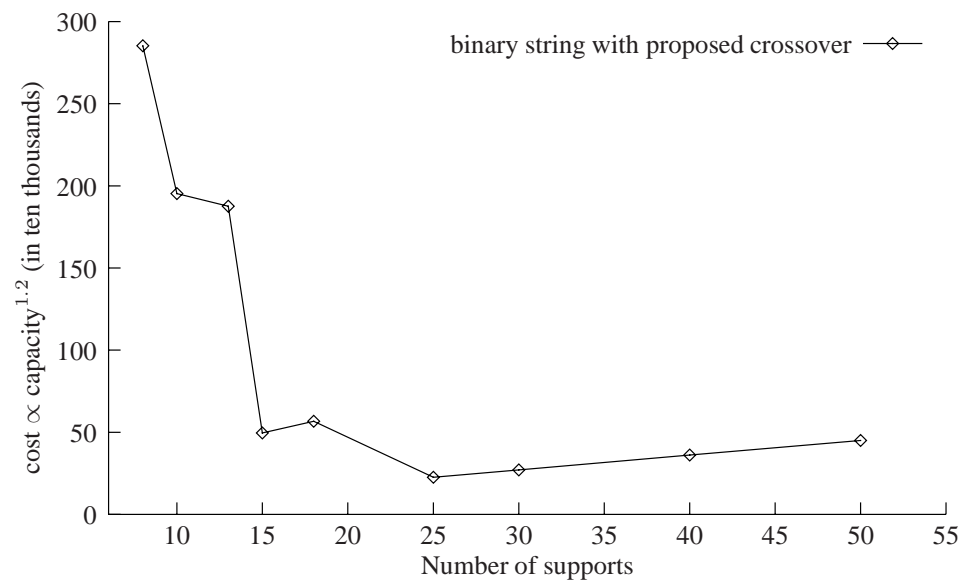


Figure 3.18: Multi-objective study of lifetime cost for the real piping system

Chapter 4

Other Results

4.1 Integer string representation

The string representation initially used in the GA is the integer string representation. In this scenario, the support size is chosen at each support location. A support of zero capacity is used to allow locations without supports. It should be noted that this representation must be used for cases where the stiffnesses of the supports vary with the support capacity. The performance of this representation with respect to lifetime cost is shown in Figure 4.1. It can be observed that the binary representation employing uniform crossover and the proposed crossover perform much better than the integer string representation. This difference is mainly due to the reduction in search space size from $(m + 1)^n$ to 2^n and $nC_{n_{req}}$ for binary string with uniform crossover and proposed crossover respectively.

4.2 Cumulative capacity optimization

It is observed in section 3.4 while discussing capital cost optimization that optimizing for number of supports with integer string representation will not yield the cost optimal solution because several different feasible solutions, with different support types but equal number of supports would exist, each having a different cost . To avoid the above problem in an integer string representation, Chiba et al. [7] also minimize the cumulative spring stiffness of the supports. Another purpose of doing so is to consider minimization for maintenance cost which is typically considered as proportional to the support capacities. Therefore, the relationship between the maintenance cost and the number of supports can be illustrated using a trade-off curve between cumulative capacity and the number of supports. The objective function in this case is the same as that given by Equation 3.9 with the exception that c_i in Equation 3.11 represents the support capacity and not the capital cost. As shown in Figure 4.2, minimization for cumulative support capacity also results in minimization for total number of supports.

4.3 Comparison of selection techniques

When using tournament selection in the GA, there are indications of the presence of premature convergence which eliminated the population diversity established by seeding. Stochastic universal selection has a minimal spread and zero bias. Therefore, this procedure is compared with tournament selection. Even stochastic sampling can introduce premature

convergence, if the fitness of the best solution is much higher than the average fitness of the population. In order to avoid this problem, scaling is done by adding a large number to the denominator of the fitness function given by Equation 3.9, so that the best fitness is not very far from the average fitness. The results generated using tournament selection and stochastic universal selection, when the GA is run using different seeds, are compared with “real” optimal solutions obtained using rigorous enumeration of the search space. From Figure 4.3, which shows the comparison for capital cost, it can be observed that both the selection procedures do equally well. But in the case of lifetime cost, shown in Figure 4.4, it is observed that stochastic universal selection marginally outperforms tournament selection.

4.4 Uniform crossover

4.4.1 Lifetime cost optimization

The optimal solutions with respect to lifetime cost generated using uniform crossover and binary representation are given in Figure 4.5. The (θ_k/θ_{kc}) values for these solutions are shown in Figure 4.7. Similarly, the optimal solutions with respect to capital cost and their (θ_k/θ_{kc}) values are shown in Figures 4.6 and 4.8 respectively. The Figures 4.7 and 4.8 show that the supports are non-uniformly loaded as the ratios are close to unity for some supports but much less than unity for the others. This behavior is thought to be an outcome of the absence of a parameter controlling the (θ_k/θ_{kc}) in the fitness function. This

resulted in developing the parameter r_{max} , the maximum value of support load to support capacity ratio in the solution. A more uniform distribution of support loads can be achieved by minimizing r_{max} , which is defined as,

$$r_{max} = \max_{1 \leq k \leq n_{sup}} \left(\frac{\theta_k}{\theta_{kc}} \right) \quad (4.1)$$

4.4.2 Modified fitness function

The lumped mass parameter, U , described in section 3.5 is used in conjunction with r_{max} to generate optimal solutions with respect to capital cost, which have the heavier masses supported as well as a low r_{max} . The fitness function is given by:

$$\max \left\{ \frac{1}{a\Delta\delta + b\Delta\sigma + c\Delta\theta + dC + fU + gr_{max}} \right\} \quad (4.2)$$

It is similar to Equation 3.14 except for using capital cost instead of lifetime cost and the absence of the $|n_{sup} - n_{req}|$ term. Alternatives to the optimal solution are generated using the topology metric and constraints only on the cost and the fitness as given by Equations 3.15 and 3.16. The optimal solution and the alternatives generated using this fitness function are given in Figure 4.9. The cost, U and r_{max} for these solutions are listed in Table 4.1. It is observed that the even though the cost of the alternatives are within 20% of the cost of the optimal, the cost optimal solution for $n_{req} = 3$ shown in Figure 3.6 has a much lower cost than this optimal solution. Thus, as seen from section 3.5 for lifetime cost, the modified fitness function does not appear to work well for capital cost too.

4.4.3 MGA solutions

MGA is used to generate alternatives to the optimal solutions with respect to lifetime cost. To begin with, in addition to the constraints given by Equations 3.15, 3.16 and 3.17, a constraint on r_{max} is also imposed for the alternative, given as,

$$r_{max} \leq r_{all} \quad (4.3)$$

It is observed that there are very few alternatives that satisfy all the four constraints. Moreover, the observation that the optimal solutions with respect to lifetime cost obtained by proposed crossover are better than those obtained by uniform crossover, and that the support load to capacity ratios for these solutions as seen from Figure 3.10 are, in general, uniform over all the supports, lower the need of the constraint in the MGA. Thus, the constraint given by Equation 4.3 is not used to define the decision space for the alternatives. The solutions generated using MGA for $n_{req} = 8$ is shown in Figure 4.10. The important aspects of the alternatives are listed in Table 4.2. It is be observed that the topology of all the alternatives bear some similarity and are not very good as the decision maker could have arrived at most of the alternatives by hand-tweaking any one of the alternatives.

4.5 Application to real life piping system

While applying the developed DSS to a real piping system the following strange behavior is seen. Even though the actual piping system required 15 snubbers, the GA is able to find a solution with just one snubber. This solution satisfied all the constraints - stress, dis-

placement and support capacity, that are considered in the model. On careful examination of the solution, it is observed that the piping system is very flexible with its fundamental frequency very close to 1 Hz. Such a low fundamental frequency for the piping system is undesirable due to in-plant low frequency operating loads. In order to overcome this problem, it is essential to have enough supports to bring the fundamental frequency greater than 5 Hz. It is found by trial and error that if the number of supports is near 15, the frequencies of the system are above 5 Hz. This is the reason for the trade-off curve for the real system shown in Chapter 3 being evaluated in the neighborhood of $n_{req} = 15$.

Table 4.1: Properties of solutions given in Figure 4.9

Solution	Cost (in thousands)	U	$\left(\frac{\theta_k}{\theta_{kc}}\right)_{max}$
Best	215.500	0.159	0.751
MGA1	246.500	0.139	0.800
MGA2	180.000	0.301	0.906
MGA3	227.500	0.175	0.823

Table 4.2: Properties of solutions given in Figure 4.10

Solution	Cost (in ten thousands)	U	$\left(\frac{\theta_k}{\theta_{kc}}\right)_{max}$
1	172.2904	0.818	0.995
2	196.4135	0.944	0.995
3	198.5535	1.050	0.975

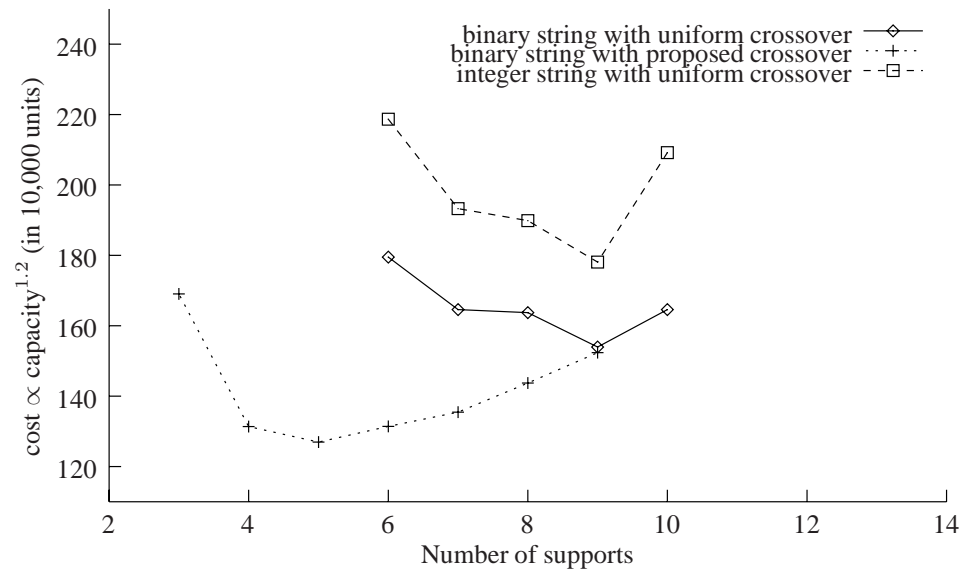


Figure 4.1: Comparison of performance of various crossover schemes for lifetime cost

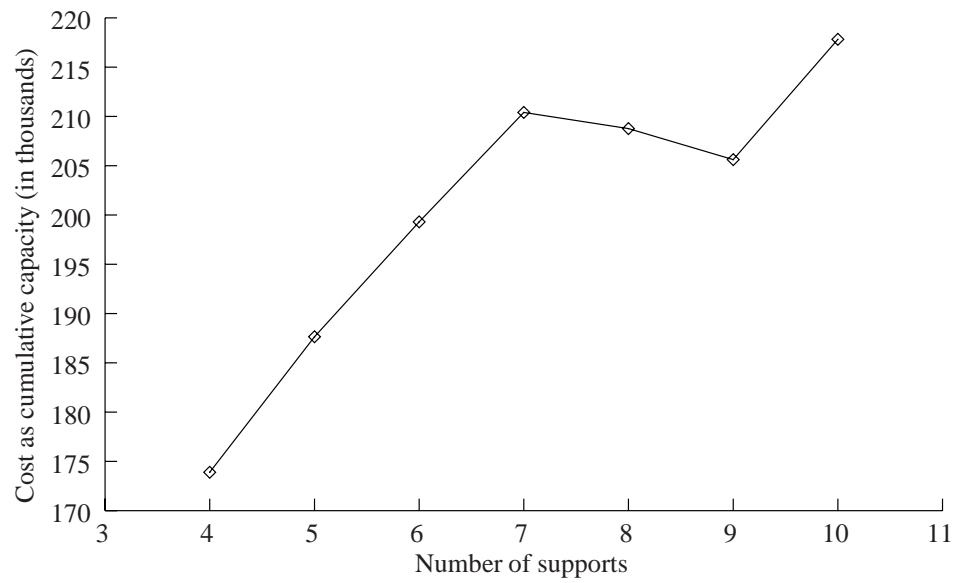


Figure 4.2: Multi-objective study cumulative capacity as cost

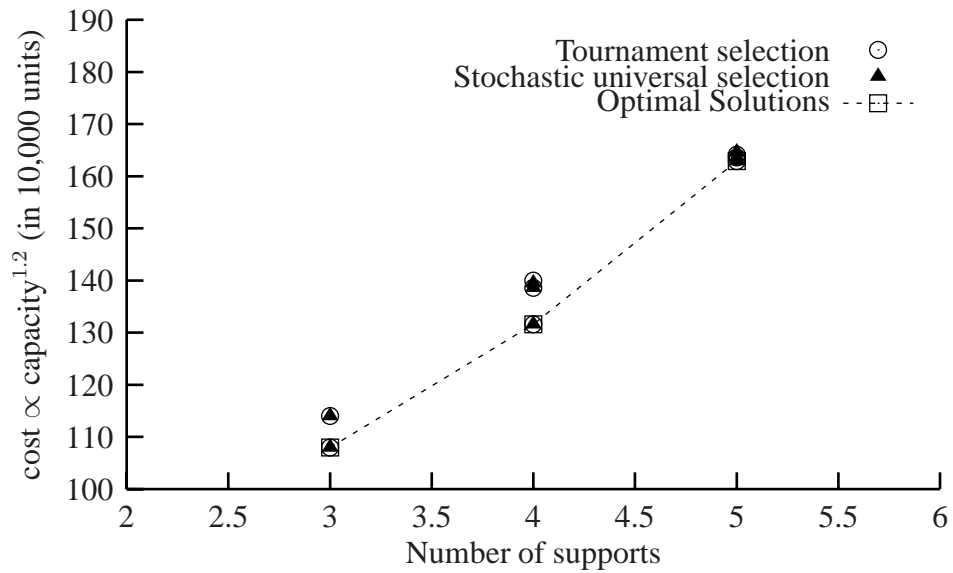


Figure 4.3: Comparison of selection schemes for capital cost using multiple seeds in GA

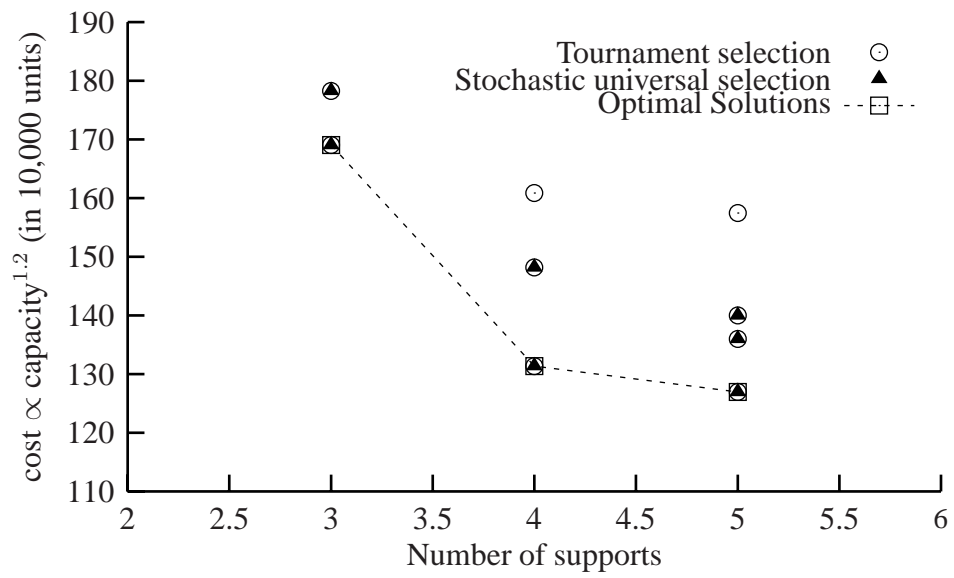


Figure 4.4: Comparison of selection schemes for lifetime cost using multiple seeds in GA

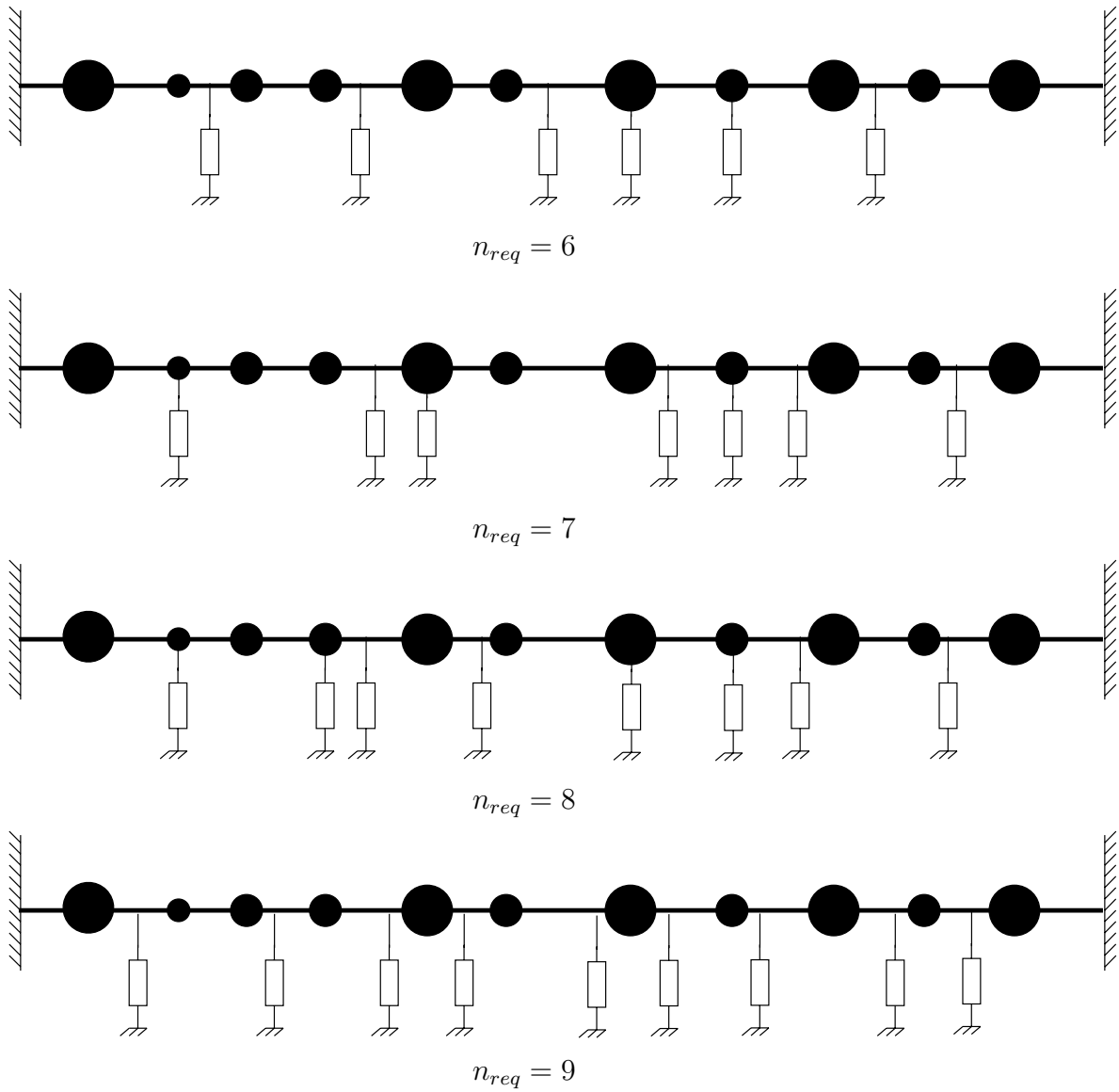


Figure 4.5: Optimal lifetime cost solutions obtained using uniform crossover and binary representation

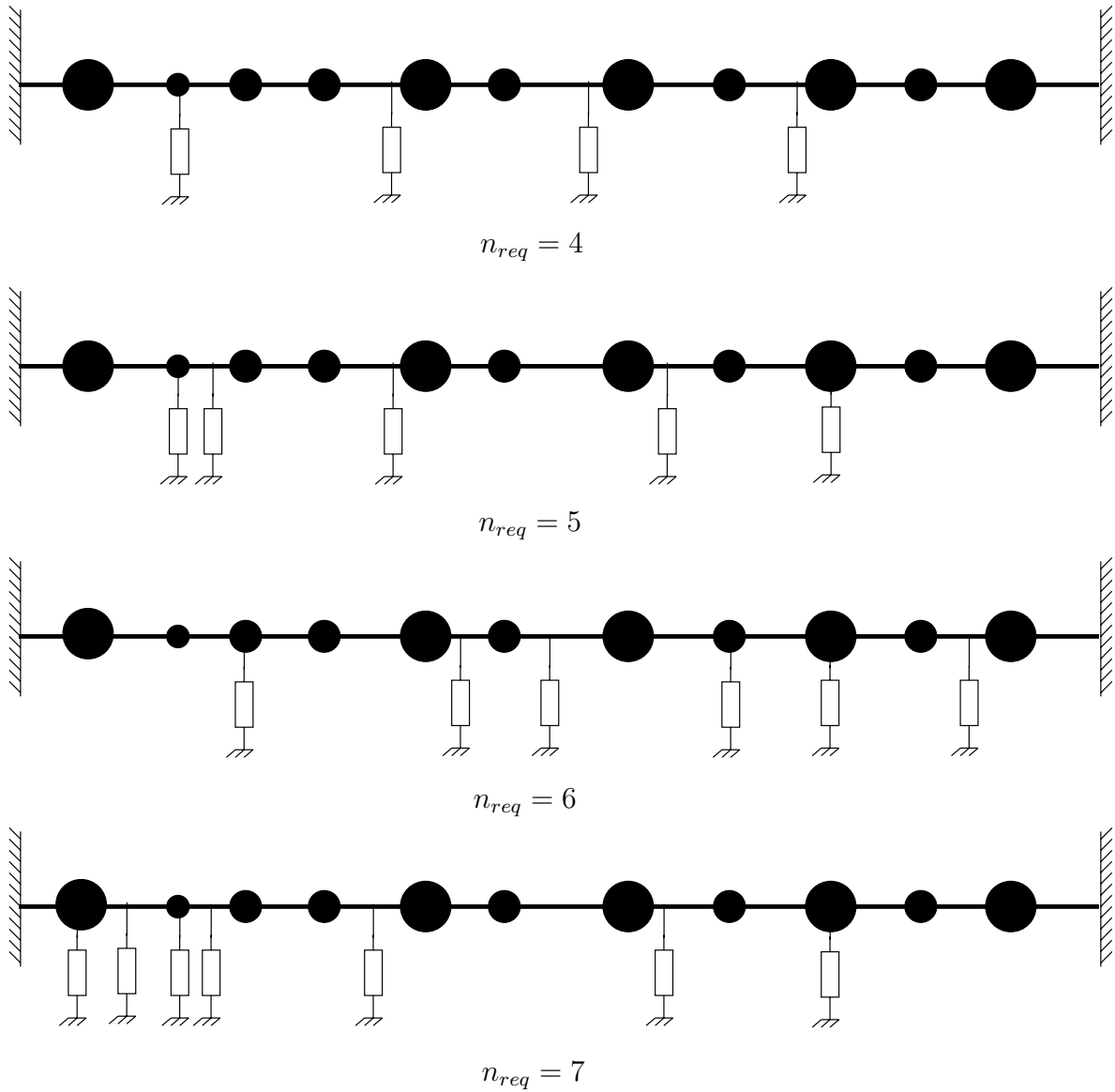


Figure 4.6: Optimal capital cost solutions obtained using uniform crossover and binary representation

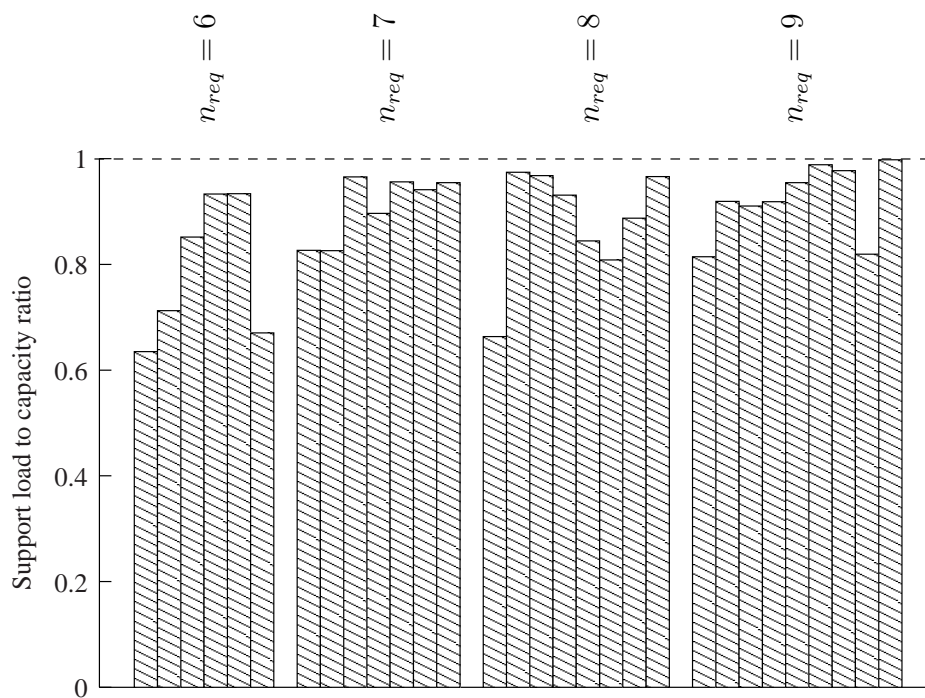


Figure 4.7: θ_k/θ_{kc} values for solutions given in Figure 4.5

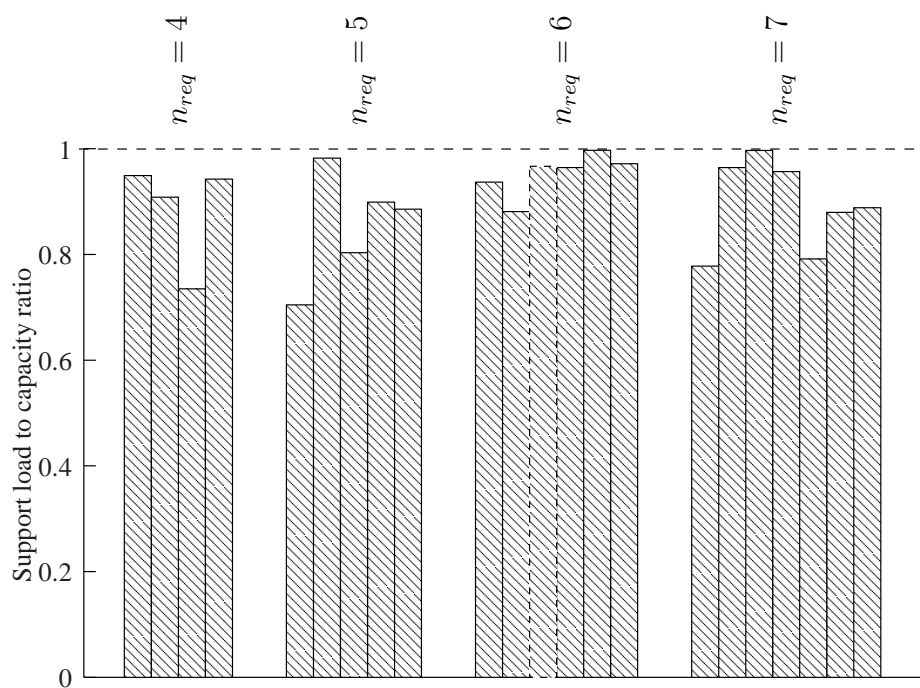


Figure 4.8: θ_k/θ_{kc} values for solutions given in Figure 4.6

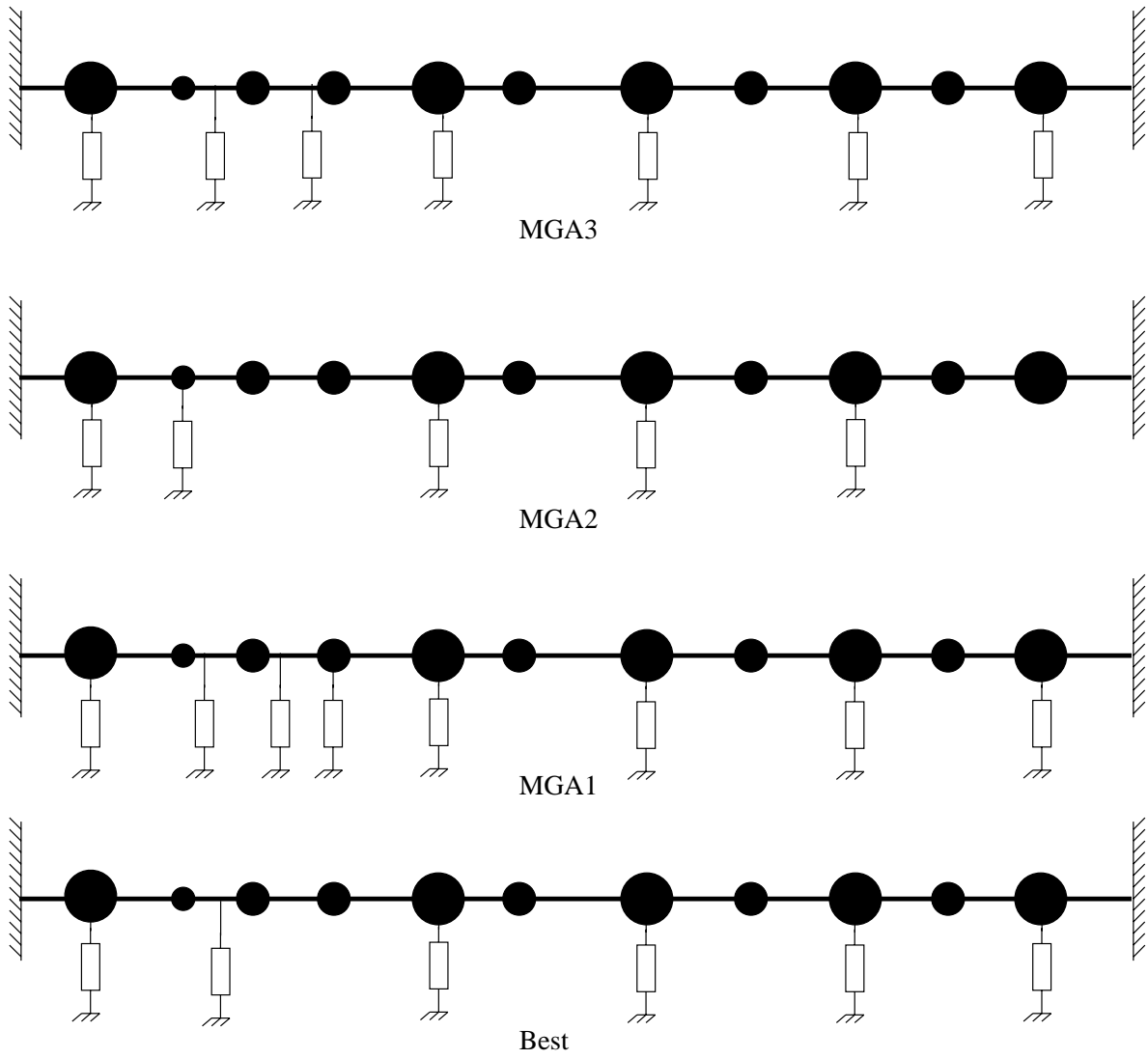


Figure 4.9: The best solution and the alternatives using Equation 4.2

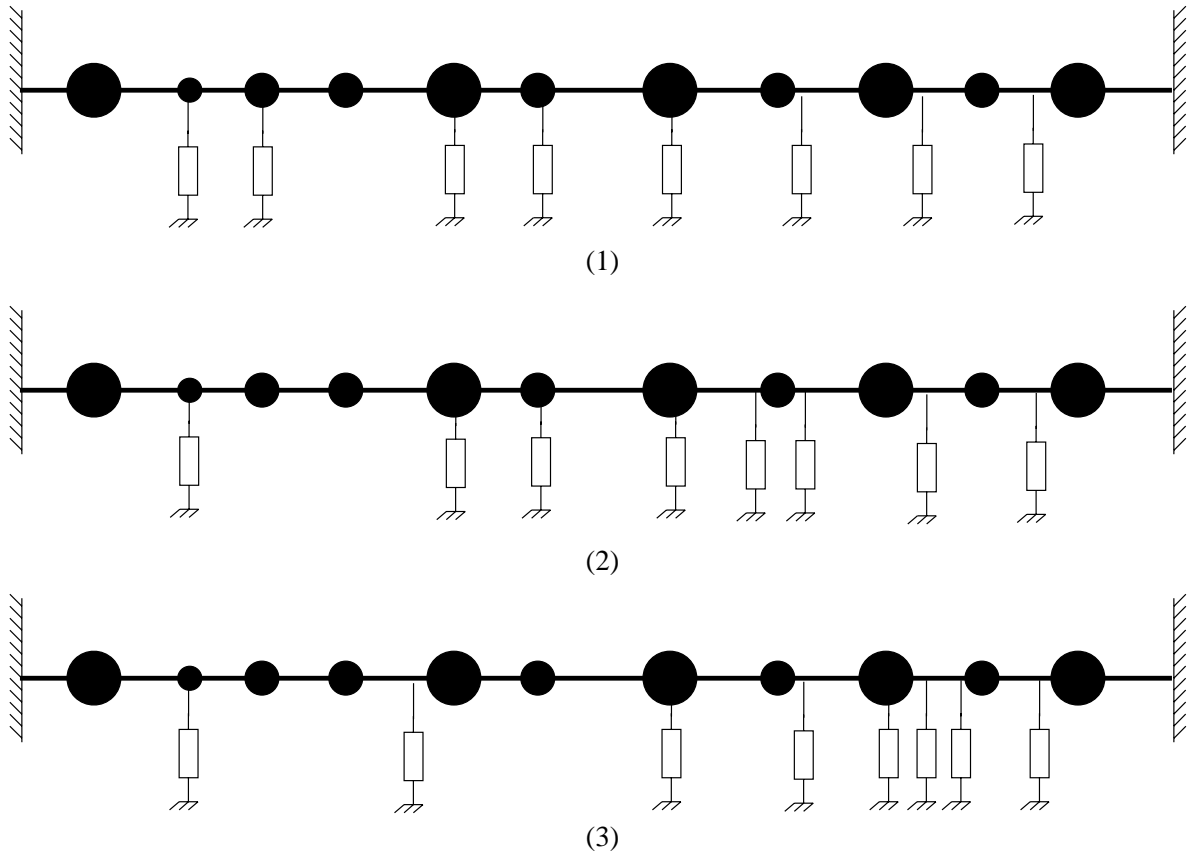


Figure 4.10: Alternatives generated for lifetime cost optimal solution of $n_{req} = 8$

Chapter 5

Summary and Conclusions

5.1 Summary

The first step in the creation of the DSS for seismic design and performance evaluation of piping supports is the implementation of a GA to optimize support locations of piping systems subject to earthquake load. Chiba et al. [7] have used integer string representation in their GA formulation that minimizes the number of supports as well as cumulative capacity of the supports. The following limitations are identified in this formulation:

- The supports are assumed to behave as springs with stiffnesses that depend on the capacity of the support, whereas in practice, supports have been observed to fail in buckling, exhibiting more of a rigid behavior.
- It is not clear whether minimizing the number of supports and their cumulative capacity minimizes the total capital or lifetime cost of the supports.

- The formulation does not address issues such as the idea of providing a support directly under heavy equipment. The engineer is likely to prefer solutions that consider these issues even if they are somewhat costlier than the optimal solution.

The above limitations are eliminated by introducing various modifications. Supports are modeled as rigid, resulting in all the supports having the same stiffness. This lead to the introduction of the binary string representation. This change is shown to produce a great reduction in the search space to be explored by the GA. Then, it is observed that seeding the GA by rolling a dice produced an initial population in which solutions have supports at 50% of the possible locations even though only a very few supports are required. Therefore, alternative seeding techniques are explored in which the population is seeded with solutions having a number of supports near the required number.

The possibility of trade-offs between various types of cost (capital or lifetime) and the number of supports is examined. The performance of the two representations are tested with respect to lifetime cost. In the process of generation of trade-off curves, cost optimal solutions for a specified number of supports n_{req} are required. It is observed that uniform crossover produces a lot of solutions with number of supports either greater than or less than n_{req} . Consequently, it is noticed that the performance of the genetic algorithm for trade-off study can be enhanced if the crossover is made to generate offspring with exactly n_{req} supports. Thus, a new crossover scheme is proposed to take care of this issue and its performance is compared with that of uniform crossover for capital cost and lifetime cost.

In order to consider the issue of having supports directly under heavy equipment, a lumped mass parameter U is proposed. The support load to support capacity ratio in the optimal solutions is seen to vary non-uniformly in the sense that, some of the supports are fully loaded while others are partially loaded. Thus, the minimization of r_{max} , the maximum value of support load to capacity ratio of the supports in the solution, is suggested. A new fitness function that considers minimization of U and r_{max} is used in the GA. The failure to support all heavier masses with a cost close to the optimal cost, is shown to suggest the usefulness of MGA. The topology metric, which is essentially the “Hamming distance,” is used as a difference metric for generating alternatives that are attractive with respect to U and r_{max} . Finally, the resulting DSS is evaluated on a real piping system.

5.2 Conclusions

The following conclusions are obtained from the process of development of the DSS for seismic design and performance evaluation of piping supports:

- Solutions are primarily infeasible with respect to support capacities. The constraints on stress and displacement are trivially satisfied in many cases.
- Binary string representation performs better than integer string representation because of the reduced search space but can be used only if all supports are modeled as rigid springs.
- The proposed crossover scheme is better suited over uniform crossover when optimal

solutions are desired for a specified number of supports n_{req} as it significantly reduces the search space.

- Seeding techniques significantly affects the performance of the GA. To generate trade-off curves, it is essential to seed the GA with solutions having exactly the required number of supports, n_{req} , to converge accurately on the optimal solution.
- Optimization for capital cost is same as minimizing the number of supports whereas optimization for lifetime cost is a multi-objective problem.
- Stochastic selection appears to work better than tournament selection for the optimization of lifetime cost while both performed equally well with respect to the optimization of capital cost.
- It might not be practically feasible to support all the heavy lumped masses of a piping system because of the high cost involved.
- MGA, using topology metric for difference and the proposed lumped mass parameter as a constraint to reduce the decision space, appears to work well.

5.3 Future work

- Explore the role of local search techniques to improve the GA solutions as several GA runs are required to find the absolute optimal.

- In the case of the trade-off study, the search space is seen to increase with the specified number of supports n_{req} . Hence, the population in the GA has to be increased accordingly. For this purpose, a mathematical relation has to be derived between n_{req} and the population size.
- The results show that the cost incurred increases with decrease in the value of the lumped mass parameter U . A trade-off study between unattractiveness and cost might be useful to the engineer in decision making.
- The excitation through all the supports is assumed to be the same whereas in reality the excitation coming through a support varies according to the floor level in the building at which the support is located. Consideration of multi-support excitation is needed to solve this problem.
- The piping system has been analyzed independently. Accounting for building-piping interactions is essential as it can further bring down the cost of the supports required.

References

- [1] 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.
- [2] J.W. Baugh Jr., G.R. Kakivaya, and J.R. Stone. Intractability of the dial-a-ride problem and a multi-objective solution using simulated annealing. *Engineering Optimization*, 30(2):91–123, 1998.
- [3] J.W. Baugh Jr. and S.V. Kumar. Asynchronous genetic algorithms using coarse-grained dataflow. In preparation, 2002.
- [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.-Y. Chang, E.D. Brill Jr., and L.D. Hopkins. Efficient random generation of feasible alternatives: a land-use example. *Journal of Regional Science*, 22(3):303–313, 1982.

- [6] S.-Y. Chang, E.D. Brill Jr., and L.D. Hopkins. Use of mathematical models to generate alternative solutions to water resource planning problems. *Water Resources Research*, 18(1):58–64, 1982.
- [7] T. Chiba, S. Okado, I. Fujii, and K. Itami. Optimum support arrangement of piping systems using genetic algorithm. *Journal of Pressure Vessel Technology*, 118:507–512, November 1996.
- [8] DST. *DST/ PIPESTRESS User's Manual*. DST Computers Services of Geneva, Switzerland, version 3.4 edition, 1992.
- [9] 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. American Society of Civil Engineers, 1991.
- [10] D.E. Goldberg. *Genetic algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing Company, 1989.
- [11] R.T. Haftka and Z. Gurdal. *Elements of structural optimization*. Kluwer Academic Publishers, Boston, third edition, 1992.
- [12] H.M. Karandikar and F. Mistree. The integration of information from design and manufacture through decision support problems. *Applied Mechanics Reviews*, 44(10), 1991.

- [13] U. Kirsch. *Structural optimization: fundamentals and applications*. Springer-Verlag, New York, 1993.
- [14] S.V. Kumar. *Vitri - A Generic Framework for Engineering Decision Support Systems on Heterogeneous Computer Networks*. PhD thesis, North Carolina State University, Raleigh, NC, 2002.
- [15] D.H. Loughlin, J.K. Neal, S. Ranjithan, E.D. Brill, and J.W. Baugh. Decision support system for air quality management. In *Computing in Civil Engineering: Proceedings of the 2nd Congress, ASCE*, pages 1367–1374, 1995.
- [16] D.H. Loughlin, S. Ranjithan, J.W. Baugh Jr., and E.D. Brill Jr. Application of genetic algorithms for the design of ozone control strategies. *Journal of the Air and Waste Management Association*, 2000.
- [17] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, third, revised and extended edition, 1996.
- [18] 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.
- [19] H.A. Simon. Problem formulation and alternative generation in the decision making process. Technical Report AIP-43, Departments of Computer Science and Psychology, Carnegie Mellon University, Pittsburgh, PA, 1988.

- [20] M.S. Voss and C.M. Foley. Evolutionary algorithm for structural optimization. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO)*, 1999.
- [21] 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.