Qualitative & Semi-Quantitative Reasoning Techniques for Engineering Projects at Conceptual Stage

M. H. Gedig
M.A.Sc, Chief Structural Engineer
AMEC Dynamic Structures1515 Kingsway Port Coquitlam, B.C., Canada V3C 1S2

and

Dr. S. F. Stiemer
Dr.-Ing, P.Eng. Professor, Department of Civil Engineering
University of British Columbia 2324 Main Mall Vancouver, B.C., Canada V6T 1Z4
E-mail: *sigi@civil.ubc.ca*

ABSTRACT

During the development of engineering projects, the level of uncertainty is not static. The level of uncertainty typically diminishes from the early, conceptual stages of the project to the latter, detailed stages. At the present time there are many tools available to the engineer for reasoning with relatively low levels of uncertainty. Unfortunately there are few resources available for drawing sound conclusions from information that is characterized by a high level of uncertainty. Since decisions made early in the project cycle generally have a greater financial impact than those made later, it is worthwhile to investigate tools which are able to provide systematic and logical evaluation of preliminary or conceptual designs. This paper investigates sound techniques for evaluating projects at the early stages, including qualitative reasoning and semi-quantitative reasoning. The paper shows that qualitative analysis methods enable the engineer to reason with a high level of abstraction. As a normal engineering project progresses, more numeric information becomes available, and the results of semi-quantitative reasoning become more useful.

KEYWORDS

Design uncertainty, conceptual design, decision evaluation, qualitative reasoning, semi-quantitative reasoning, interval technique.

1 Introduction

1.1 The Development of an Engineering Project

In a typical scenario in an engineering office, a customer provides a set of functional specifications. The engineering firm develops a preliminary design to suit these specifications. Based on the preliminary design, the engineering firm and customer agree to a budget estimate and schedule. The engineering firm then develop the preliminary design to the final, detailed design.

1.2 Design Constraints

The functional constraints provided by the client are a subset of a number of explicit and implicit constraints on an engineering project. In addition to functional constraints, the engineer must abide by cost or budgetary restrictions, physical laws that govern the behaviour of all physical systems, and practical considerations, which relate to matters such as fabrication, shipping, installation and maintenance.

1.3 Sources of Information

As a project progresses, new information continually becomes available. As a result, the level of uncertainty changes during the life of the project. As the engineer looks with greater detail into each area of the project, questions are raised, which may require the client to further refine the scope of the project. Also, the engineer may discover further, unforeseen limitations of the design. As the engineer develops and tests analysis models, he gains a greater understanding of the behaviour of the system under study.

1.4 Design Changes

Usually, the engineer begins by developing several conceptual designs and chooses the best alternative following a preliminary analysis. In the ideal scenario, he would simply add detail to the chosen conceptual design to arrive at the final design. As the project progresses, changes to the overall concept become less and less practical. Later details become constrained by earlier details. Given normal time and economic constraints, it is not practical to modify the concept as soon as a detail does not fit. The early conceptual decisions have greater impact on the status of the project than the latter details.

1.5 Current Analysis Tools

There are currently many numerical analysis tools that are available for the detailed analysis of engineering models. Unfortunately, most of these tools require that the input parameters for the model be completely specified and organized to fit within a rigid format. Although these tools are unquestionably invaluable during the latter stages of a project, there is some question as to their value during the early stages of a project. Numerical analysis tools may be used during the early stages, but will generally require simplified models and rough approximations. Success with a primitive model does not guarantee that a more detailed, accurate model will verify the design. In addition, with most numerical analysis tools, the uncertainty in the results is not calculated or related to the uncertainty of the input parameters.

1.6 A Proposed Analysis Tool

A desirable objective for an analysis tool would be to rule out unacceptable conceptual designs at the earliest possible stage. At the early project stages, the definition of “unacceptable” is unclear because of the relatively high level of uncertainty. The goal of this research is to develop a tool to evaluate conceptual designs using proven, logical, and sound reasoning techniques. The tool must be able to cope with a high level of uncertainty, and flexible enough to accept more information as the project progresses. An analysis tool, QES, was developed as part of this research; this tool is discussed in greater detail in a follow-up paper. The goal of this paper is to investigate the reasoning techniques that may be useful at the early stages of an engineering project. These techniques include qualitative and semi-quantitative reasoning.

2 Qualitative Reasoning

2.1 Overview

Qualitative analysis has been used for years in a diverse number of fields, including economics, social sciences, mathematics and engineering. In recent years, qualitative analysis has been used in a field of artificial intelligence called qualitative reasoning about physical systems, or qualitative physics. Formal representation schemes have been developed which may be used to reason qualitatively about the behaviour of physical systems. Qualitative physics provides a formal, systematic approach to qualitative analysis, and constitutes a key part of the qualitative reasoning application that was developed as a part of this research. This discussion will focus on the topic of qualitative dynamics, which deals with the behaviour of physical systems over time. There are other areas of qualitative reasoning, such as qualitative spatial reasoning, which is of interest in robotics, but this topic involves a more complex range of issues than need be described here.

2.2 Qualitative Physics

Qualitative physics is an area of artificial intelligence concerned with reasoning qualitatively about the behaviour of physical systems [10]. Physical systems include any systems, both natural and fabricated, which operate under the laws of physics. Qualitative physics evolved from the study of “commonsense” physical reasoning. Early in the study of artificial intelligence, researchers realize that humans are able to effectively function with seemingly little effort when faced with situations characterized with uncertainty. The field of commonsense physical reasoning, or naive physics [9], sought to formalize the types of knowledge required reason about everyday situations. Qualitative physics, while embodying the goals of commonsense reasoning, also seeks to predict the behaviour of physical artifacts, and to generate plausible explanations for their behaviour. Qualitative physics deals not only with everyday devices, but also with complex physical systems.

Human experts are able to reason with incomplete information in more complex domains. It is common in engineering, for example, for the expert to use complex qualitative reasoning strategies to answer a given question about a domain. Certainly the expert, who generally has knowledge of the underlying physical principles of the domain, can set up equations and attempt to solve them numerically or analytically. Before attempting a detailed solution, however, experts often reason qualitatively in a way that is different from commonsense reasoning.

Research in qualitative physics was spurred on by the shortcomings of expert systems, the first commercially-viable applications of artificial intelligence. Experts systems typically rely on domain-specific heuristic knowledge, and tend to fail ungracefully when confronted with problems that fall even slightly outside the domain for which the system is intended. A key fault of most expert systems is their inability to reason using fundamental knowledge of the domain, such as conservation laws. On the other hand, qualitative reasoning uses a detailed domain model, an explicit representation of the first principles of the domain that underlie heuristic knowledge. Expert systems are often characterized by shallow knowledge; domain knowledge that has been encoded in a format that is particularly efficient for a specific application. Qualitative reasoning uses deep knowledge, which refers to fundamental knowledge of the domain.

A further problem of expert systems is that much of their knowledge is encoded as implicit rules. Expert systems generally include several different types of knowledge, such as experiential knowledge, rules of thumb, basic principles, and empirical knowledge. No distinction is made between these different types of knowledge, and as a result, expert systems are difficult to transfer to another domain. It is a problem to determine which types of knowledge apply only to a specific domain, and which are applicable over an number of different situations.

The need for an explicit domain model makes some domains more suitable than others for applying qualitative reasoning. The domain of physical systems, for example, is appropriate for qualitative reasoning because a set of fundamental laws of physics is available. In other domains, such as medicine and ecology, a comparable set of laws does not exist. In medicine, for example, most of the knowledge used for diagnosis and treating diseases is empirical, not based on a model of the relevant biological and chemical mechanisms.

2.3 Qualitative Calculus

Qualitative reasoning entails reasoning with information that is less precisely specified than numerical data. The signs, relative magnitude and directions of change in quantities are all types of information that can be used to reason qualitatively. A language has been developed in which it is possible to express imprecise knowledge of quantities and the relations between quantities. The terms *qualitative variables* and *qualitative equations* are used to describe concepts which are analogous to concepts in algebra. The value of a usual algebraic variable specifies only one element in the set of possible values. If the set is the real number line, a value represents a specific point. On the other hand, the value of a qualitative variable can represent one element as well as a set of elements.

Functional relationships between quantities may also be described qualitatively. For example, take Hooke's law of ideal springs, which is expressed as

$$F = kx$$

where F is the force applied to the spring, k is the spring constant, and x represents a distance from an equilibrium position. This relation can be expressed qualitatively as: "The greater the force on the spring, the larger the displacement from the equilibrium position."

Qualitative Variables

Qualitative variables may be constructed from continuous variables. The entire domain of a continuous variable is subdivided into a finite number of nonoverlapping subintervals, and all the values in the same interval are treated as equivalent. The qualitative value of a variable is the name of the interval in which the actual numerical value lies.

For example, a qualitative variable which represents the temperature of water might consist of three intervals: $(-\infty, 0)$, the temperature in degrees Celsius over which water exists in the solid form; $(0, 100)$, where water is a liquid; and $(100, +\infty)$, in which water is a vapour. It is important that the continuous domain is subdivided into subintervals which represent qualitatively distinct regions of behavior. The representation scheme should capture distinction that make an important qualitative difference, ignoring others. As an example, if the intervals $(0, 80)$ and $(80, 100)$ were used instead of $(0, 100)$ to represent the temperature of water, this would have introduced unnecessary complexity into the model, which may have an adverse effect on the solution process.

The boundary values used to subdivide a continuous domain are called *landmark values*, or *distinguished values*. For the variable representing water temperature, the landmark values are at 0 degrees and 100 degrees. A variable can have any number of landmark values at which behaviour changes significantly, but qualitative reasoning is most efficient when their number is finite. Using a finite number of landmarks supports case analysis and reasoning by exclusion, two powerful techniques of qualitative reasoning.

The most common subdivision of continuous domains is into three subintervals, labeled *negative*, *zero*, and *positive*. These three subintervals represent the negative numbers, zero and the positive numbers. This subdivision will be used to explain some of the general concepts of the qualitative calculus. One significant aspect of the division into negative numbers, zero, and positive numbers is that when a variable represents the rate of change of the value of some quantity, the value of the qualitative variable indicates whether the quantity is decreasing, constant, or increasing. Representations using more landmark values have been developed; some of these will be described later in the context of specific applications of qualitative reasoning.

The *domain of signs* is the set of qualitative values which results when the landmark zero (0) is used, dividing the positive numbers from the negative numbers on the real number line. The domain of signs is expressed as

$S \in \{ +, 0, - \}$.

It is also useful to use the set of *extended signs*, *S'*, written as

$S' \in \{ +, 0, -, ? \}$,

which allows us to express ignorance of the sign of a quantity.

The fundamental definition of the domain of signs is given in Table 1 which defines the intervals which are associated with the elements of the set.

Table 1: The domain of signs.

Interval	Sign
$(0, +\infty)$	+
$[0, 0]$	0
$(-\infty, 0)$	-
$(-\infty, +\infty)$	?

A number of sign-valued operators are defined, which when applied to real numbers, return signs as values:

$$[X]_0 = \text{sign}(X) = \begin{cases} + & \text{if } X > 0 \\ 0 & \text{if } X = 0 \\ - & \text{if } X < 0 \end{cases} \tag{1}$$

More generally, $[X]_{X0} = \text{sign}(X - X_0)$, where X_0 serves as a reference for the variable *X*. The operation $[X]_0$ is often abbreviated by $[X]$. In addition, assuming that the variable *X* is continuous and differentiable everywhere,

$$[\dot{X}] = [dX/dt]_0 = \text{sign}(dX/dt) \tag{1.2}$$

Table 2: Qualitative addition [*X*] + [*Y*].

+	[*X*]			
	+	0	-	?

	+	+	+	?	?
[*Y*]	0	+	0	-	?
	-	?	-	-	?
	?	?	?	?	?

Addition over the real numbers is a function that takes two real values and returns a third. It is not possible to define addition over signs because the sum of the qualitative values + and - is ambiguous (it can be either +, 0, or -). Addition can be represented as a function when defined over the extended signs however, as shown in Table 2.

For convenience, multiplication and the unary negation operator are also defined over the extended signs, as presented in Table 3 and Table 4.

Table 3: Qualitative multiplication [*X*] × [*Y*].

×		[*X*]			
		+	0	-	?
	+	+	0	-	?
[*Y*]	0	0	0	0	0
	-	-	0	+	?
	?	?	0	?	?

Table 4: Qualitative negation -[X].

		-[*X*]
	+	-
[*X*]	0	0
	-	+
	?	?

Kuipers [11] represents addition and multiplication as relations instead of functions. When qualitative addition and multiplication are viewed as functions, uncertainty can propagate through a chain of calculations. For example, if we wish to evaluate the expression [*Z*] × ([*X*] + [*Y*]) when the value of [*X*] is + and the value of [*Y*] is -, the expression in brackets evaluates to ?, so the result of the multiplication operation, and also, any other expressions that depend on this one, is ?. When addition is viewed as a relation, three signs are evaluated and a value of *true* (T) or *false* (F) is returned. Evaluating the expression amounts to asking the question: "is it true or false that adding [*X*] to [*Y*] will result in the value + / 0 / - ?" Although viewing addition as a relation gives us no more information than when it is viewed as a function, it helps to prevent the spread of uncertainty.

2.4 Qualitative Equations

The qualitative addition and multiplication operators and relations act as qualitative constraints on the values of variables. Qualitative equations are derived from models of physical systems in much the same way as the usual quantitative equations. In fact, quantitative equations are often converted into qualitative equations. In general, a quantitative equation can be transformed into a qualitative equation using rules of the form shown in Table 5.

As an example of a qualitative equation, consider Newton’s second law of motion, $F = m{\cdot}a$, which governs the relationship between the resultant force acting on a body of mass m and the acceleration a of the body. Since the mass is positive and constant, $[m] = +$ and $[\dot{m}] = 0$. We can transform this equation into a qualitative equation and draw several useful conclusions:

Force and acceleration are in the same direction:

$$[F] = [m{\cdot}a] = [m][a] = [a].$$

A change in either force or acceleration results in a change to the other in the same direction:

$$[\dot{F}] = [d(ma)/dt] = [m][\dot{a}] + [a][\dot{m}] = [\dot{a}].$$

Table 5: Rules used to generate qualitative equations.

$[e_1+e_2] \Rightarrow [e_1] + [e_2]$	$[e_1{\cdot}e_2] \Rightarrow [e_1][e_2]$
$[0] + [e] \Rightarrow [e]$	$[0]{\cdot}[e] \Rightarrow [0]$
$[+][e] \Rightarrow [e]$	$[-][e] \Rightarrow -[e]$

Qualitative equations may also be derived from qualitative knowledge of how variable values depend on others. For example, we may notice that the viscosity of a liquid decreases as its temperature increases, although we might not know the precise functional relationship between viscosity and temperature. The relation is expressed qualitatively as

$$[\dot{v}] = -[\dot{k}],$$

where v and k represent viscosity and temperature, respectively.

A quantitative equation can also be transformed into a qualitative one by linearizing the equation using a Taylor series expansion [4].

2.5 Solution of Qualitative Equations

Solving systems of qualitative equations means finding qualitative values of variables that satisfy the equations. Unlike systems of equations in real variables, a system of ‘independent’ qualitative equations in general does not have a unique solution, even when the number of variables is the same as the number of equations. The problem of finding a set of qualitative variables consistent with a given set of qualitative equations is a constraint-satisfaction problem where the equations are the constraints to be satisfied. Techniques for solving constraint-satisfaction problems can be applied to systems of qualitative equations.

A technique that is commonly used to find the values of the variables is local constraint propagation. Given the values of all but one of the variables in an equation, the value of the unknown may be determined by substituting in the values of the known variables and simplifying the expression. This process may be viewed as propagating values through networks of equations. As an illustration of local propagation, consider the following system of equations:

$$[a] + [b] = [c] \quad (2)$$

$$[d] = [a] \quad (3)$$

$$[e] + [f] = [b] \quad (4)$$

Given the values $[d] = 0$, $[e] = +$, $[f] = 0$, equation (4) simplifies to $[b] = +$. Since $[d]$ is zero, equation (3) implies that $[a]$ must be zero also. Substituting these two results into the first equation and simplifying yields $[c] = +$.

Sets of qualitative equations are generally inherently simultaneous, which means they cannot be solved by substitution alone. The usual algebraic techniques also do not work. For example, consider the following system of equations:

$$\begin{aligned} x + y &= 3 \\ 2x + y &= 4 \end{aligned} \quad (5)$$

There is no sequence of simple propagation which can solve this problem. Such inherent simultaneity cannot be resolved, say, by subtracting $x + y = 3$ from $2x + y = 4$, yielding $x = 1$. This strategy fails in the qualitative domain because the addition and multiplication operators defined for qualitative values do not have associated with them the field axioms which are required to perform this type of manipulation. One of the problems is that sign algebra lacks an additive inverse, so that inferences such as

$$[s] = [t] \Rightarrow [s] + [u] = [t] + [u], \text{ and}$$

$$[s] + [t] = [u] \Rightarrow [s] = [t] - [u],$$

in general, are not valid with qualitative variables [21].

Local constraint propagation methods often do not work, and even though they are able to find a solution, they cannot guarantee that they will find all results. In general a set of qualitative equations cannot be solved to obtain a unique solution. One of the reasons for this is that qualitative algebra is inherently ambiguous, as indicated by the table for qualitative addition (Table 2). If the qualitative variable $[f]$ from the previous example had been assigned the negative sign, $[f] = -$, substituting this value and $[e] = +$ into equation (4) the value of $[b]$ is undetermined, or $[b] = ?$. Any of the three signs +, 0 or - is a possible assignment for $[b]$. Three new cases, corresponding to $[b] = +$, $[b] = -$, and $[b] = 0$, are generated in order to continue the propagation. This ambiguity causes a very large number of possible cases to be generated as the system becomes large.

In general, local propagation is possible when the qualitative equations take the following form:

$$X_1 = c,\ X_2 = f_1(X_1),\ X_3 = f_2(X_1, X_2),\ \ldots$$

where c is a constant, and $f_1, f_2, \ldots$ represent explicit expressions that can be evaluated. When there are simultaneous equations such as

$$X_1 = g_1(X_1, X_2),\ X_2 = g_2(X_1, X_2),$$

then more powerful constraint-satisfaction methods are required.

Another approach to constraint satisfaction is to use a generate-and-test approach, where all possible combinations of assignments to variables are made. Each possible set of assignments is checked against the equations in order to determine if the assignment implies a contradiction. This method is guaranteed to find all possible solutions to the system of equations, however it is inefficient. More efficient means of obtaining all solutions exist. Some of these methods are discussed in the following section, which deals with constraint-satisfaction methods.

Systems of qualitative equations have multiple solutions because the field axioms of qualitative algebra are weaker than their quantitative counterparts. The ambiguity of qualitative calculus causes the search for a solution to grow very rapidly with the number of variables in the system and the number of possible assumptions. There have been several suggestions for dealing with the problems of ambiguity and search complexity in qualitative calculus. Most involve using additional knowledge, including quantitative knowledge.

Search efficiency can be improved by using domain-specific and problem-specific knowledge. For example, if we are studying the thermodynamics of a kettle on a stove, we may justifiably ignore the loss of heat from the kettle to the air. Allowing qualitative variables a greater number of possible values is another approach to resolving ambiguity in qualitative analysis. The representation developed by Kuipers in QSIM [11], discussed later, illustrates one possibility for using a richer set of qualitative information. Some of the ambiguities may be removed using knowledge about ordinal relations between variables. An arithmetic reasoning system called Quantity Lattice [18], is capable of representing and reasoning with information on partial ordering relations amoung variables and expressions. Finally, Dormoy and Raiman [5] developed a technique for solving qualitative equations in a manner analogous to the Gaussian elimination procedure that is used to solve sets of conventional linear equations.

It should be noted that multiple solutions are not just an artifact of qualitative analysis. They are an important part of the theory, since each solution describes a behaviour that can potentially occur in the operation of the system being modeled. Since the qualitative model is an abstraction of the physical device being modeled, it is possible that the model may represent some other realizable device. Thus qualitative analysis can be useful in design applications, because it is capable of finding solutions to systems of constraints which are not obvious, but nonetheless valid.

3 Constraint-Satisfaction Methods

Constraint-based reasoning is a model for formulating knowledge as a set of constraints without specifying the method by which these constraints are to be satisfied. Many problems are more naturally expressed in terms of what is allowed or, conversely, what is not allowed. A large number of the current computational systems, however, insist on a rigid separation between input and output information. Constraint-based reasoning is becoming more prevalent as a means of flexibly representing a range of problems. A variety of techniques have been developed for finding partial or complete solutions for different kinds of constraint expressions.

3.1 Constraint-Satisfaction Problems

Constraint-satisfaction problems (CSPs) follow a general formulation, which consists of a finite set V of n variables $\{V_1, V_2, \ldots, V_n\}$, each associated with a domain of possible values, D_1, D_2, ... D_n and a set of constraints $\{C_1, C_2, \ldots, C_t\}$. Each of the constraints is expressed as a relation defined on some subset of variables, given that there are subsets of the Cartesian product of the domains of the variables involved. The set of solutions is the largest subset of the Cartesian product of all the given variable domains such that each n-tuple in that set satisfies all the given constraint relations.

An assignment of a unique domain value to each member of some subset of variables is called an instantiation. An instantiation is said to satisfy a given constraint C_i if the partial assignment specified by the instantiation does not violate C_i. An instantiation is *legal* or *locally consistent* if it satisfies all the constraints in which it is involved. A number of different tasks may be defined in connection with this formulation. Typical objectives are to determine whether a solution exists (the *decision problem*), to find one or all of the solutions, or to determine whether an instantiation of some subset of the variables is a partial solution.

Several restrictions on the general definition of a CSP are possible. The domains may be required to have a finite number of discrete values. It may be required that all relations are unary or binary, that is, that they only constrain individual variables or pairs of variables, respectively.

The techniques used in solving constraint-satisfaction problems can be classified into three categories [3]. The first category consists of search techniques which are used to systematically explore the space of all solutions. The most common algorithm in this class is *backtracking*, which traverses the search space in a depth-first fashion. The second category is *consistency algorithms*, which are generally used to perform preprocessing in order to improve the efficiency of the backtracking search, but can also be incorporated in the search. The third technique, *structure-driven algorithms*, can be used to support both the consistency algorithms and the backtrack search. Structure-driven algorithms will not be discussed here.

A technique exists to solve any constraint-satisfaction problem. Assuming finite discrete domains, the assignment space $D = D_1 \times D_2 \times \cdot \cdot \cdot \times D_3$ is finite, so one may evaluate each constraint on each element of D. Once a legal instantiation is found the problem is solved. This generate-and-test algorithm is correct but slow. A more intelligent technique, backtracking, may be used to "prune" away significant portions of the assignment space.

3.2 Backtracking

Backtracking algorithms systematically explore the assignment space D by sequentially instantiating the variables in some order. As soon as any constraint has all its variables instantiated, its truth value is determined. If the constraint is not satisfied, that partial assignment cannot be part of any total valid assignment. Backtracking then falls back to the last variable with unassigned values remaining in its domain (if any).

As an example of backtracking search, consider three variables V_1, V_2, and V_3. The only valid assignment to each of these variables is one of the integers 1, 2, or 3, so that the domains of these variables may be specified as $D_1 = D_2 = D_3 = \{1, 2, 3\}$. The constraints for this problem are

$C_1(X)$:	$X > 1$
$C_{12}(X_1, X_2)$:	$X_1 > X_2$
$C_{23}(X_1, X_2)$:	$X_1 > X_2$

where C_i indicates a unary constraint on the variable V_i, and C_{ij} represents a binary constraint on the variables V_i and V_j. The constraints take as arguments domain elements of the appropriate variable, and return a truth value. For example, $C_{12}(2,1)$ evaluates to the truth value *true* (T).

The results of the backtracking search are given in the search tree in Table 6.

3.3 Constraint Networks

In discussing constraint-satisfaction problems, it is convenient to view the task specification as a network which consists of a labeled, directed graph. In the graph, variables are represented by nodes, each with an associated set representing the domain of the variable, unary constraints are represented by loops on the nodes and the binary constraints by labeled, directed arcs between nodes. This type of network is called a binary constraint network. The constraint graph for the network associated with the backtracking example problem is shown in Figure 1. In this graph, the arcs for $C_{ij}^{T} = C_{ji}$ are not shown.

Table 6: Backtracking search tree.

V_1	V_1V_2	$V_1V_2V_3$	
1			
2			
	2 1		
		2 1 1	
		2 1 2	
		2 1 3	
	2 2		
	2 3		
3			
	3 1		
		3 1 1	
		3 1 2	
		3 1 3	
	3 2		
		3 2 1	•
		3 2 2	
		3 2 3	
	3 3		

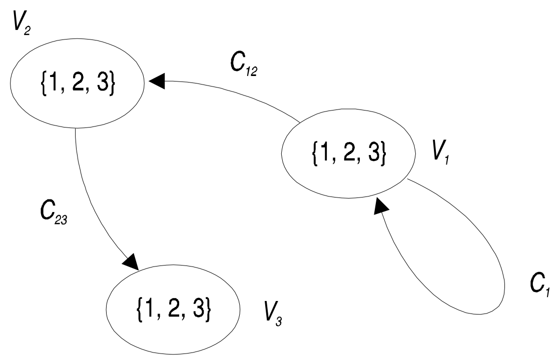


Figure 1: Constraint graph for example problem.

3.4 Consistency Techniques

At times, backtracking can be grossly inefficient. A variety of consistency-enforcing algorithms have been developed which may be performed prior to the search in order to improve its efficiency Montanari [14] Mackworth [13] Freuder [7]. These algorithms transform a given constraint network into an equivalent, yet more explicit network by deducing new constraints to be added on to the network.

The most obvious source of inefficiency in backtracking and the easiest to prevent concerns the unary constraints. If the domain D_i for a variable V_i includes a value that does not satisfy $C_i(x)$ then it will be the cause of repeated instantiation and failure. Algorithms which enforce *node consistency* remove the cause of this repeated failure by simply discarding all those domain elements which do not satisfy the corresponding unary constraint. The term node consistency is used because unary constraints are represented as loops on nodes in the constraint graph, as shown by constraint C_1 in the example problem. The example problem is made node consistent by removing the element 1 from the domain of variable V_1.

Another source of inefficiency in backtracking search is called *arc inconsistency*. An arc from node V_i to V_j is inconsistent if there exists a value a in the domain of V_i such that C_{ij} does not hold for any value of V_j when $V_i = a$. In the example problem, the constraint network is arc inconsistent. For example, when $V_3 = 2$, there exists no value of the variable V_2 such that constraint C_{23} is satisfied. Inconsistency may be removed from an arc by comparing the two nodes joined by the arc, and deleting those elements which cause inconsistency. Removing an element from a node, however, may allow further deletions from the nodes adjacent to the modified node. For this reason a single pass over the network will not guarantee that the network is arc consistent. The process of enforcing arc consistency between two nodes must be repeated until there is no reduction in any domain. Waltz developed a method of achieving arc consistency in one pass through the nodes by using information about the arcs leading into and out of a node [20]. Mackworth developed a more elegant version of the Waltz algorithm which, however, does not attempt to make the network consistent on a single pass [13].

A generalization of arc consistency techniques is to path consistency [14, 13]. A further generalization to p-ary relations is the concept of k-consistency ($1 \le p, k \le n$) developed by Freuder. A network is k-consistent if and only if, given any instantiation of any k-1 variables satisfying all the direct constraints amoung those variables, it is possible to find an instantiation of any kth variable such that the k values taken together satisfy all the constraints amoung the k variables. Node, arc and path consistency correspond to k-consistency for $k = 1$, 2, and 3, respectively.

Freuder uses an alternate form of constraint network, where constraints are represented by nodes instead of arcs. The procedure for achieving k-consistency begins by building a constraint network consisting of all the unary constraints. Next, nodes are added representing the binary constraints. A binary constraint is synthesized from each pair of unary constraints and two arcs are added, from the binary constraint to the two unary constraints from which the binary constraint was synthesized. The next step is to add a node representing a ternary constraint (a constraint involving three variables). Ternary constraints may be synthesized from the binary constraints. Arcs are then added from the ternary constraint to the appropriate binary constraints. In an incremental approach, new constraints are continually synthesized from lower order constraints. After a new constraint has been created and the network augmented, it is propagated along arcs to the neighbouring constraints. Synthesized constraints represent constraint networks on a subset of the variables in the problem. The subset of variables grows as the procedure continues, until it includes the entire set of variables in the problem. At this point the solution to the problem is the last constraint synthesized.

By successively adding higher level nodes to the network and propagating constraints in the augmented network, it is possible to achieve k-consistency for all k. It is not necessary, as with the arc consistency algorithms discussed earlier, to restrict the given constraints to binary relations. The synthesis algorithm reins in combinatorial explosion by progressively ruling out lower order inconsistencies in stages. As mentioned, the algorithm can be used to find explicitly all the solutions to the constraint-satisfaction problem. In practice, it has been found more efficient to run the algorithm for a few steps as a preprocessor to simplify subsequent backtracking search.

4 Example of Qualitative Analysis

An example of qualitative analysis from the field of structural mechanics is given in this section. This example illustrates some of the practical implications of qualitative analysis, as well as some refinements that may be made.

The pin-jointed plane structure model shown in Figure 2 will be studied using qualitative analysis techniques. In the figure, the boxed numbers are the element labels and the other numbers are the node labels. The parameters of this model are the lengths (L_1, L_2, L_3, L_4) and axial stiffnesses (EA_1, EA_2, EA_3, EA_4) of each of the four members, as well as the nodal displacements at the upper left node (Δ_{X2}, Δ_{Y2}), and the upper right node (Δ_{X3}, Δ_{Y3}). The horizontal displacements Δ_{X2} and Δ_{X3} are considered positive if to the right, and the vertical displacements Δ_{Y2} and Δ_{Y3} are positive if upwards. A horizontal force which acts to the right is applied at the upper left node. One of the uses for such a model would be to estimate the deflected shape of the structure. This is the objective of the qualitative analysis.

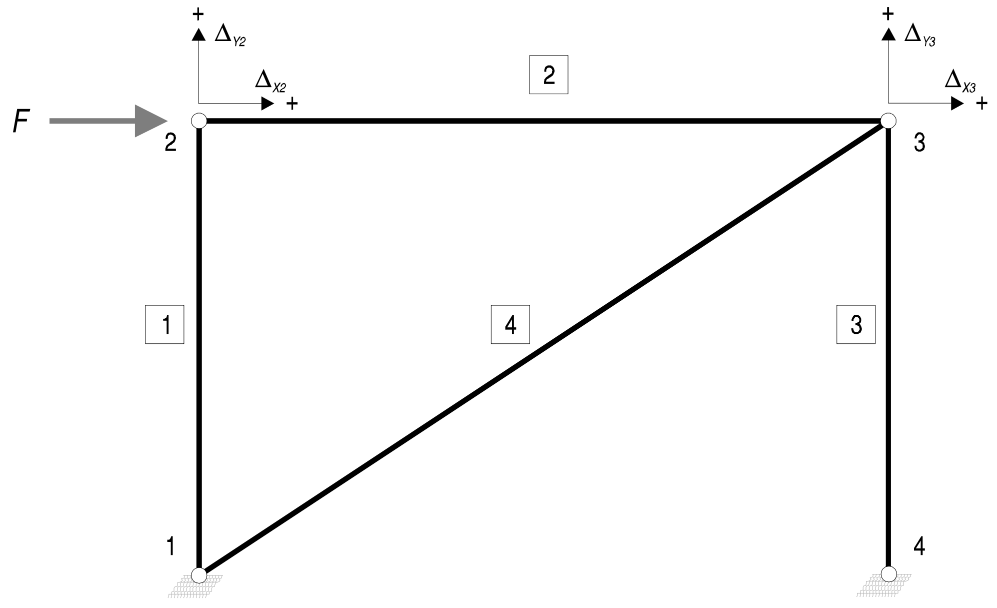


Figure 2: Pin-jointed structure.

The *qualitative* equations which relate the nodal displacements to the material properties and geometry of the structure are given as follows:

$$\frac{E_2A_2}{L_2}[\Delta_{X2} - \Delta_{X3}] - F = 0 \tag{6}$$

$$\frac{E_1A_1}{L_1}\Delta_{Y2} = 0 \tag{7}$$

$$\frac{E_2A_2}{L_2}[\Delta_{X3} - \Delta_{X2}] + \frac{E_4A_4}{L_4}[\Delta_{X3} + \Delta_{Y3}] = 0 \tag{8}$$

$$\frac{E_3A_3}{L_3}\Delta_{Y3} + \frac{E_4A_4}{L_4}[\Delta_{X3} + \Delta_{Y3}] = 0 \tag{9}$$

These qualitative equations are very similar to the usual quantitative equations. The difference is that positive numeric constants have been omitted. Positive constants may be eliminated from products, because the positive sign value acts as the identity for multiplication in the domain of signs.

Qualitative analysis produces four solutions to the problem (Table 7), the last of which is the correct solution. The result of this analysis is a typical result in qualitative analysis: multiple solutions are generated but the ‘correct’ solution to the problem is always contained in the set of solutions. The other solutions are not, strictly speaking, incorrect, because they are a correct solutions to the *qualitative* model. A qualitative model is usually a generalization of an associated quantitative model. The process of generalizing a quantitative model allows solutions which do not satisfy the quantitative equations. One of the prime sources of weakness in qualitative predictions derives from the weakness of the qualitative addition function. Looking at equation (6), the assignment Δ_{X2} = -, Δ_{X3} = - causes the term (Δ_{X2} - Δ_{X3}) to evaluate to ?. Since the sign ? represents the interval [-∞, +∞], this combination of assignments satisfies equation (6), even though such values would not satisfy the corresponding qualitative equation. As mentioned previously, the qualitative addition function introduces uncertainty which tends to propagate through systems of qualitative equations.

Table 7. Results of Qualitative Analysis of Pin-Jointed Structure

Solution	Δ_{X2}	Δ_{Y2}	Δ_{X3}	Δ_{Y3}
1	+	0	-	+
2	-	0	+	-
3	0	0	+	-
4	+	0	+	-

One way of strengthening the conclusions drawn by qualitative analysis is to make some use of ordinal relations between qualitative variables [6]. Ordinal relations may be used to reduce the uncertainty caused by the qualitative addition operator. Whenever qualitative addition results in the sign ?, three new cases may be generated to reflect different possible ordinal relations between the arguments involved in the addition operation (Table 8).

Table 8: Qualitative addition with ordinal relations.

+		[X]		
		+	0	-
[Y]	+	+	+	- |X| > |Y|; 0 |X| = |Y|; + |X| < |Y|
	0	+	0	-
	-	- |X| < |Y|; 0 |X| = |Y|; + |X| > |Y|	-	-

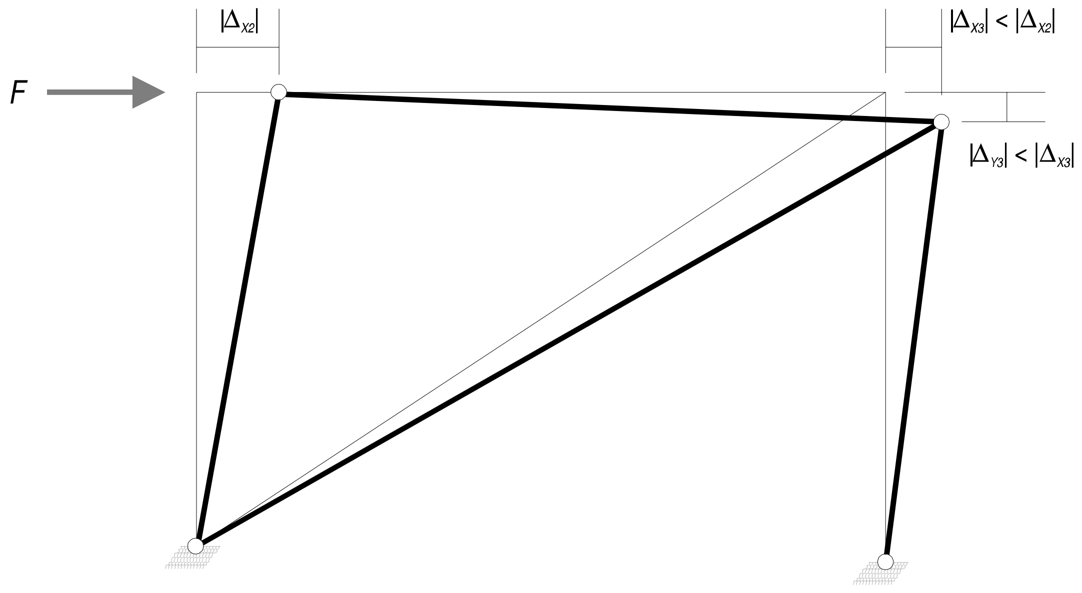
Since qualitative analysis is troubled by combinatorial explosion, at first glance it may seem that considering ordinal relations between variables would exacerbate this problem. Without including ordinal relations, the number of combinations of two variables, each having three values, is nine, while there are thirteen possible combinations when ordinal relations are used. The advantage in using ordinal relations is that the relations provide additional information that may be used to filter inconsistent variable assignments. The ordinal relations are assertions about the relationship between two variables. This assertion may cause a contradiction at some stage of the solution process, which allows us to delete elements from the partial solution.

In addition to information about the values of the variables, a solution obtained using ordinal relations provides insight into the relationship between the magnitudes of variables. The analysis yields the following additional information:

$|\Delta_{X3}| > |\Delta_{Y3}|$

$|\Delta_{X2}| > |\Delta_{X3}|$

These inferred ordinal relations enable us to generate a fairly concise qualitative graphical representation of the behaviour of the structural model, as shown in Figure 3.



$|\Delta_{X2}|$
$|\Delta_{X3}| < |\Delta_{X2}|$
F
$|\Delta_{Y3}| < |\Delta_{X3}|$

Figure 3: Graphical representation of solution to pin-jointed structural problem.

In this example, qualitative analysis leads to a fairly clean, informative solution, considering the minimal amount of specific information that was furnished as input. It is important to note that, in general, a qualitative analysis produces more than one solution. Increasing the number of variables in a problem leads to a rapid increase in the number of combinations which must be considered, and also, an increase in the number of solutions.

In the previous example from structural analysis, very little information was specified about the various quantities in the problems. In most problems in engineering, partial knowledge about quantities takes the form of partial numerical values. For example, the value of Young's Modulus E was only specified as lying in the interval $(0, +\infty]$. In reality, we may know that the structure is to be constructed of timber, so that the value of E ranges, say, from 6000 MPa for sawn timber to 14000 MPa for glued-laminated timber. Even if we have almost no idea of which material is to be used, a wide interval which includes values of E for timber, concrete and steel will provide some information. In this general case, E would range from about 6000 MPa for timber to 200000 MPa for steel, so that the range of E could be represented as the interval [6000, 200000] MPa. This observation suggests that, for an engineering application at least, it is more beneficial to pursue a formulation involving partial numeric information, rather than to seek refinements to the pure qualitative representation.

5 Semi-Quantitative Reasoning

5.1 Overview

The engineer is rarely confronted with a situation where purely qualitative information is available. Almost invariably there exists some knowledge of certain quantities, even though these may be incomplete or partially-specified. In order to make full use of incomplete information, it is necessary to be able to reason in some way with partial numeric data. Semi-quantitative reasoning is the task of combining incomplete quantitative and qualitative knowledge. Semi-quantitative reasoning is important to model-based reasoning tasks such as design, monitoring and diagnosis. All of these tasks involve incomplete knowledge in both qualitative and quantitative forms.

There are a number of different representations available for reasoning with incomplete knowledge of quantities, including bounding intervals, probability distribution functions, fuzzy sets, and order-of-magnitude relations. This thesis focuses on the use of bounding intervals to represent partial knowledge of a real number. The discipline of interval analysis provides a rich array of techniques for working with intervals [01, 15, 16, 17].

5.2 Interval Analysis

Interval analysis was originally developed as a means of providing rigorous error bounds on the results of machine computations. Since all computers perform arithmetic using numbers that are represented with finite precision, it is of interest to know how far a computed result would lie from the result calculated using real numbers. An interval arithmetic was developed which employs a special type of rounding that guarantees an interval which contains both ordinary machine arithmetic results and infinite precision arithmetic results. Interval arithmetic is not just used as a means of studying error bounds on machine computations, however. Interval techniques provide important methods with which one can reason on the range of values of variables.

Interval analysis is now a well-developed branch of applied mathematics. Interval computations are used in a wide range of applications where numeric uncertainty is concerned. For example, in performing a cost-benefit analysis of a project, the World Bank measures the project's profitability by computing the rate-of-return of a projected cost-benefit stream [16]. The rate-of-return is a root of a polynomial whose coefficients are the projected cost-benefit stream. Interval methods have been used by the World Bank to find upper and lower bounds on the rates-of-return.

Since interval methods provide a method for dealing with uncertainty, interval analysis includes techniques which are useful in engineering. Engineers use mathematical equations to model the behaviour of physical systems. Even if it is possible to solve the mathematical equation exactly, which is not usually the case, the result is still only an approximate description of the behaviour of the real system. Mathematical models used in engineering often contain constants which are determined experimentally by measurements. The numerical values assigned to such constants will therefore have limited precision. Using interval methods one is able to compute bounds on the set of solutions corresponding to intervals of possible values for the measured quantities.

An interval of real numbers can be thought of as a pair of real numbers, representing the endpoints of the interval. When the endpoints are the same, the interval is said to be *degenerate*. Degenerate intervals represent real numbers, so that interval arithmetic includes real arithmetic as a special case. This distinction is somewhat meaningless, because it is generally impossible to perform real arithmetic. Most arithmetic is performed on computers, which are confined to approximate arithmetic of limited precision.

An interval is defined as a closed bounded set of real numbers:

$$[a, b] = \{ x : a \le x \le b \}$$

An interval can also be regarded as a number represented by the ordered pair of its endpoints *a* and *b*, just as a rational number *a*/*b* is represented by an ordered pair of integers and a complex number $x + iy$ by an ordered pair of real numbers.

The interval representation is well-understood, widely-used, and quite powerful. Many researchers in artificial intelligence have developed and applied methods for combined algebraic, ordinal, and interval reasoning. Although simple, there are a number of important benefits to using the interval representation. Its simplicity means that descriptions of incomplete knowledge of quantities in the form of bounding intervals are easy to obtain. Intervals are also flexible enough to represent both an approximate value and a degree of uncertainty. Intervals can be compactly represented, using two real numbers and two booleans (to represent whether the interval is closed, open, or half-open). All connected sets of intervals are intervals (and vice versa). The intersection of two intervals is an interval, so the consequence of two interval descriptions of the same quantity is easy to compute by intersecting the intervals.

One concern which arises with regard to the interval representation is whether it is reasonable to treat quantitative knowledge as a step function, to simply state that a quantity lies between two values and nothing more. It may seem more intuitive to think in terms of probability distributions, with a likelihood that gradually diminishes on either side of a central value. The alternatives to the interval representation for representing uncertainty are generally more complex [2]. Probability functions require more arbitrary assumptions than simple intervals, their semantics are less clear, and if done properly, require more complex computations. Representing uncertain values by, say, a Gaussian probability curve is no less arbitrary than representing them by fixed bounds, and it can be difficult to define what the probability means and how to combine different curves. In the context of engineering decision-making, decisions are sometimes made on the basis of subjective assessments of probability distributions. Research has shown that biases are prevalent in such assessments, leading to significant errors [19]. One of the objectives of this thesis is to employ methods which enable one to reason with less information. Probability functions require more information to describe parameters than the usual numeric techniques, and so are not suited to the type of semi-quantitative reasoning discussed here.

5.3 Solution of Interval Equations

The interval representation is a flexible and simple framework which can be used to represent partial knowledge about quantities. Although interval arithmetic has many similarities with the traditional arithmetic involving real numbers, there are a number of complexities which are introduced in dealing with intervals rather than with real numbers.

Consider the linear system of algebraic equations

$$A_{11}x_1 + A_{12}x_2 = B_1 \tag{9}$$

$$A_{21}x_1 + A_{22}x_2 = B_2, \tag{10}$$

where the coefficients A_{11}, A_{12}, A_{21}, A_{22}, as well as B_1 and B_2, are *intervals*:

$$A_{11} = [2, 3] \qquad A_{12} = [0, 1]$$

$$A_{21} = [1, 2] \qquad A_{22} = [2, 3]$$

$$B_1 = [0, 120] \qquad B_2 = [60, 240]. \tag{11}$$

In the first quadrant, where

$$x_1 \geq 0 \text{ and } x_2 \geq 0, \tag{12}$$

the members on the left of (9) and (10) may be written as $[2x_1, 3x_1 + x_2]$ and $[x_1 + 2x_2, 2x_1 + 3x_2]$, respectively. Since the intersections

$$[2x_1, 3x_1 + x_2] \cap [0, 120], \text{ and}$$

$$[x_1 + 2x_2, 2x_1 + 3x_2] \cap [60, 240]$$

must not be empty, the following inequalities result:

$$2x_1 \leq 120; \qquad 3x_1 + x_2 \geq 0;$$

$$x_1 + 2x_2 \leq 240; \qquad 2x_1 + 3x_2 \geq 60.$$

In this two-dimensional example, the solution may be represented in the first quadrant as a polygon with vertices (30, 0), (60, 0), (60, 90), (0, 120), and (0, 20). If this process is repeated for the remaining three quadrants, we find that the solution is a polygon in the second and fourth quadrants as well, but that no part of the solution lies in the third quadrant. Combining these results, the solution to the equations is an eight-sided polygon (Figure 4). Even in this two-dimensional case, the solution set is not particularly easy to represent. In higher-dimensional cases, the difficulty is obviously compounded.

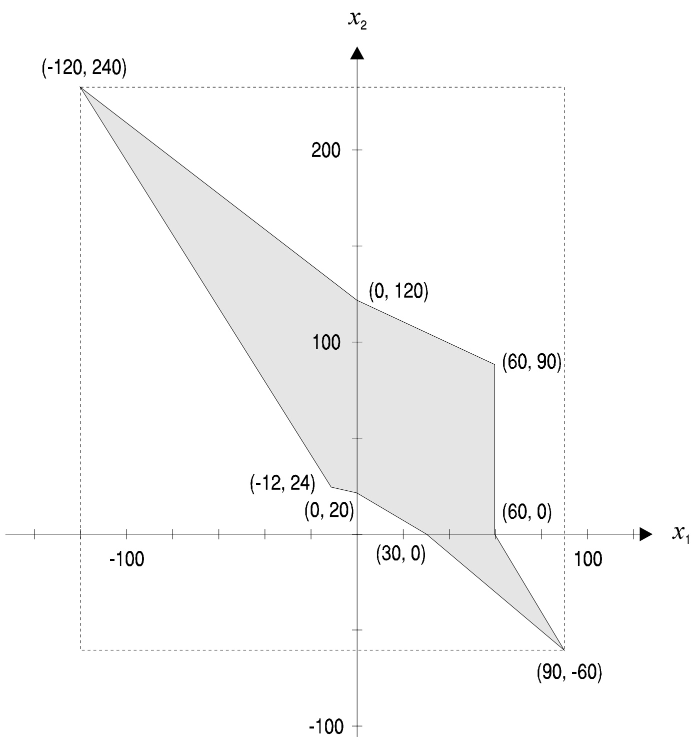


Figure 4: Solution set for set of linear equations with interval coefficients.

5.4 Numeric Constraint-Satisfaction Problems

Systems of interval equations can be formulated as numeric constraint-satisfaction problems. As discussed earlier, a constraint-satisfaction problem is defined by a set of variables, each with an associated domain of possible values and a set of constraints on the variables. In numeric constraint-satisfaction problems the domains are continuous while the constraints are numeric relations. Since the domains are continuous, it is impossible to enumerate all of the solutions, which is feasible with finite domains. Even if the domains are finite integer domains, it is generally not possible to find all solutions, because of the large number of combinations which may be formed. This difficulty may be overcome by associating a dynamic domain with each variable and by propagating this domain through the constraints.

Constraint propagation is a technique used in enforcing consistency in networks of constraints [13]. Similar consistency techniques have been used to solve numeric constraint-satisfaction problems [2, 12]. The goal of these numeric consistency techniques is neither to enumerate all solutions nor to algebraically solve a system of constraints, but to determine the exact ranges of values of the different variables. A key concept of numeric consistency techniques is that the domains of variables are represented by intervals, and that as constraints are propagated from one variable to another, the bounds of the intervals are updated dynamically. By using established techniques for dealing with intervals, numeric consistency techniques are able to guarantee that the correct result will be bounded by the domains of each variable.

6 Conclusions

This paper has described a number of techniques which enable the engineer to evaluate physical systems which are characterized by a high level of uncertainty. The qualitative and quantitative techniques discussed here are useful in the evaluation of conceptual and preliminary engineering designs. These techniques show considerable advantages over other, more commonly used approaches for handling uncertainty in engineering practice.

6.1 Design Abstraction

An important feature of qualitative and semi-quantitative methods is that they allow the engineer the reason with a high level of abstraction. Qualitative equations are capable of representing a class of physical systems rather than one specific system. In the structural example, the corresponding quantitative equations would represent a very specific system, with one set of member lengths and section properties. On the other hand, the qualitative equations represent a large number of physical structures, with a range of materials, section properties, and member geometry.

6.2 Design Alternatives

Qualitative and semi-quantitative methods are useful for evaluating conceptual and preliminary designs, because by reasoning with an abstract model, the engineer may perform a sound preliminary evaluation of a physical system without committing to details. All system parameters can be represented using qualitative variables or interval variables that are capable of covering a wide range of possible system configurations. The overall integrity of the proposed system can be confirmed at an early stage in the design, so that sound conceptual design alternatives are retained while unsound alternatives are eliminated.

6.3 Detailed Numerical Analysis

A wide range of tools are available which are able to perform detailed numerical analysis. These tools generally require input that is specific and complete. The user must commit to a large number of detailed assumptions about the physical system. At the earliest stages of design, the validity of these detailed assumptions is questionable. A considerable amount of time may be required to generate the assumptions and create the detailed model of the system.

6.4 Other Techniques for Handling Uncertainty

Qualitative and semi-quantitative methods have distinct advantages over other approaches for dealing with uncertainty in engineering systems, such as probability methods, Monte Carlo simulation, hill-climbing techniques and expert systems.

There exist analysis tools which are based on probability methods. These techniques are particularly useful when the uncertain quantities are the result of random natural processes which can be measured and easily represented by probability distribution functions. When random natural processes are not involved, arbitrary assumptions about the shape of probability functions must be made. In comparison to quantitative and semi-quantitative approaches, probability techniques generally require that more information be specified about the quantities involved before an analysis can be made. In the evaluation of conceptual designs, an objective is to draw as many useful conclusions as possible from the limited amount of available information rather than to generate a large number of arbitrary assumptions.

Expert systems represent another approach to coping with uncertainty in engineering practice. These systems have been applied successfully in engineering domains, partly because they are able to reason with valuable experiential and heuristic knowledge. Most expert systems suffer, however, because of an inability to reason using fundamental domain knowledge. On the other hand, qualitative reasoning uses a detailed domain model, an explicit representation of the first principles of the domain that underlie heuristic knowledge.

Numerical simulation techniques based on probability, such as Monte Carlo methods, and hill-climbing techniques, which seek the maximum and minimum values of each quantity, are not sound inferential techniques. and thus may exclude legitimate solutions. They generally return a subset of the true range, and thus arbitrarily exclude legitimate possibilities. Experience with hill-climbers has found them to be slow and unreliable. The techniques discussed in this paper are logically sound. The interval analysis methods employed are able to guarantee bounds on the correct solution.

6.5 Limitations of Quantitative and Semi-Quantitative Analysis

Purely qualitative reasoning methods developed in artificial intelligence research were applied to a practical problem in the domain of structural engineering. It was found that the conclusions that may be drawn from such methods can be quite powerful when the number of variables in the problem is limited. Unfortunately the conclusions that may be drawn become rapidly weaker as the number of unknowns in the problem is increased. Although a limited number of problems were tested, the findings are not out of line with previous research. A number of attempts have been made by artificial intelligence researchers to resolve the ambiguity of predictions made using qualitative analysis. The approach used in this research was to augment qualitative analysis with partial quantitative information. Semi-quantitative techniques show considerable promise in the practice of engineering for their ability to model design abstraction while providing bounds for the behaviour of physical systems.

6.6 Further Research

In order to fully evaluate the techniques presented in this paper, the techniques must be used in real engineering applications. A software program has been developed as part of this research. The goal of developing this software was to provide an accessible tool which could be used for the analysis of conceptual and preliminary engineering designs. The implementation of qualitative and semi-quantitative techniques in software is discussed in a follow-up paper.

Further research and software development is recommended in order to establish the range of application of the analysis techniques discussed in this paper.

7 References

1. Alefeld, G., and Herzberger, J. (1983) *Introduction to Interval Computation*. Academic Press, New York.

2. Davis, E. (1987) Constraint propagation with interval labels. *Artificial Intelligence* **32**:281-331.

3. Dechter, R. (1992) Constraint networks. In Shapiro, S.C. (Ed.), *Encyclopedia of Artificial Intelligence, 2nd Ed*. John Wiley, New York.

4. De Kleer, J. and Brown, J. (1984) A qualitative physics based on confluences. *Artificial Intelligence* **24**:7-83.

5. Dormoy, J. and Raiman, O. (1988) Assembling a device. *Proc., Seventh National Conference on Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA. 95-113.

6. Forbus, K.D. (1984) Qualitative process theory. *Artificial Intelligence* **24**:85-168.

7. Freuder, E.C. (1978) Synthesizing constraint expressions. *Communications of the ACM*, **21**:958-966.

8. Gedig, M.H. (1995) *A Framework for Qualitative and Semi-Quantitative Analysis in Engineering Design and Evaluation*. Masters Thesis, University of British Columbia.

9. Hayes, P.J. (1979) The naive physics manifesto. In Michie, D. (Ed.), *Expert Systems in the Microelectronic Age*. Edinburgh University Press, Edinburgh.

10. Iwasaki, Y. (1989) Qualitative physics. In Barr, A., Cohen, P.R., and Feigenbaum, E.A. (Eds.), *The Handbook of Artificial Intelligence, Vol IV.* Addison-Wesley, Reading, MA. 323-414.

11. Kuipers, B.J. (1994) *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA.

12. Lhomme, O. (1993) Consistency techniques for numeric CSPs. In *Proc. Thirteenth Joint Conference on Artificial Intelligence, IJCAII 1993*, Morgan Kaufmann, San Mateo, CA. 232-238.

13. Mackworth, A.K. (1977) Consistency in networks of relations. *Artificial Intelligence* **8**:99-118.

14. Montanari, U. (1974) Networks of constraints: fundamental properties and applications to picture processing. *Inform. Sci.* **7**:95-132.

15. Moore, R.E. (1966) *Interval Analysis*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

16. Moore, R.E. (1979) *Methods and Applications of Interval Aritihmetic*. Studies in Applied Mathematics, SIAM, Philadelphia.

17. Neumaier, A. (1990) *Interval methods for systems of equations*. Cambridge University Press, Cambridge.

18. Simmons, R. (1986) Commonsense arithmetic reasoning. *Proc. Fifth National Conference on Artificial Intelligence* 118-124.

19. Tversky, A., and Khaneman, D. (1977) Judgement under uncertainty: heuristics and biases. In *Modern Decision Analysis*, Penguin Books, London.

20. Waltz, D.L. (1975) Understanding line drawings of scenes with shadows. In Winston, P.H. (Ed.), *The Psychology of Computer Vision* 19-92.

21. Williams, B.C. (1991) A theory of interactions: Unifying qualitative and quantitative algebraic reasoning. *Artificial Intelligence* **51**:39-94.