TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 16/67 pp96-101
Volume 13, Number S1, October 2008

Evaluation of Metal Fatigue Problems Using Qualitative Reasoning Approach

Ye Zhou†, Siegfried F. Stiemer**

† Dynamic Structures Ltd., Port Coquitlam, BC V3C1S2, Canada
Civil Engineering Department, University of British Columbia, Vancouver, BC V6T1Z4, Canada

Abstract: Fatigue failures are often encountered in steel structures under heavy cyclic loadings. This paper presents metal fatigue problems in structural engineering using outcomes of recent advancements in numerical qualitative reasoning. Qualitative reasoning provides an effective and sound technique for solving complex and uncertain scenarios, regardless of the uncertainty or linearity of the design parameters and their constraints. This paper introduces the algorithms behind a software platform, built upon numerical qualitative reasoning for engineering applications. The software expresses the results of the analysis in variable ranges and diagrams showing a two-dimensional design space. The capability of representing design parameters and outcomes in solution spaces provides a practical way for engineers to leverage their existing knowledge and experience. Case studies in metal fatigue design are given to reflect on the capability of qualitative reasoning in engineering applications.

Keywords: computer applications; conceptual design; decision support system; constraints; qualitative reasoning; fatigue

1 Qualitative Reasoning with Uncertainties

Any engineering design task can usually be decomposed into a set of relationships and constraints, which can be readily represented in terms of inequalities. Inequality constraints define solutions in the form of solution spaces. Single point solutions are sought in engineering due to the fact that complete solution spaces are too difficult to compute and manage. Using solution spaces can be extremely helpful during engineering decision-making[1].

Qualitative reasoning is capable of deriving the complete solution space from a set of constraints[2]. The key technique used in qualitative reasoning is constraint satisfaction. Engineering analytical tasks are well suited to formulation as constraint satisfaction problems (CSP), which are defined by a set of variables subject to constraints. The variables correspond to the relevant parameters of the design formulas. The CSP approach uses search methods that detect single variable assignments that satisfy all the constraints, and then provide a description of solution spaces, i.e., the set of the entire solutions. This paper summarizes a research with following introduced methods[3,4].

- The approximation of solution spaces is achieved by an improved local consistency method for numerical variables providing good results in pruning and execution time. This is achieved by using a local consistency operator for numerical constraints, which is superior in pruning power to existing methods.
- A novel search method using local consistency for numerical variables is implemented. In

Received: 2008-05-28
** To whom correspondence should be addressed.
E-mail: sigi@civil.ubc.ca

contrast to most existing approaches for solving mixed CSPs that are based on a cooperation between constraint solvers, this method integrates the local consistency methods for numerical variables into the search process, and also makes use of the mixed constraints to prune the search space.

2 System Architecture

Qualitative engineering system 2 (QES2) is a software implementation of the theories and algorithms described in the preceding section. QES2 enables an engineering design or analysis to account for uncertainties in the form of ranges of input design parameters, and illustrate analysis results in the form of plots of solution spaces. QES2 is based upon the techniques of numerical constraint reasoning, interval arithmetic (IA) and adaptive plotting.

The core component of QES2 is a solver for reasoning with numeric constraints. The reasoning solver works with a collection of interval arithmetic libraries to handle various mathematical as well as logical functions such as sin(), cos(), and iif(). The versatile input/output interface uses an input parser and tree-node representation of output information. The local consistency of the reasoning process is computed by interval arithmetic.

While the reasoning solver and its input parser and IA library provide QES2's ability to analyze problems, the post processing modules, including a design space plotter and a result set manager, allow users to effectively interpret the analysis results, visually and numerically. The design space plotter illustrates graphically the two-dimensional relationship between any selected sets of design parameters. The result set manager may be used to archive, retrieve and superimpose any sets of analysis results for various purposes such as result set comparisons. Engineering design plug-ins can be custom built to streamline input for specific engineering problems.

2.1 Constraint solver

The core of QES2 is its numeric constraint solver, consisting of Java classes UInterface, Exp, and Interpreter, as shown in Fig. 1.

UInterface is the interface between the user input and the analysis process, i.e., the constraint reasoning process, and also facilitates the data flow between all the reasoning components. All user input of variables and constraints are provided by user interface modules in the form of text strings. UInterface uses a parser to convert these input strings to Exp, a binary-tree expression, on which constraint reasoning can be employed. Upon requests from UInterface, Interpreter does the constraint reasoning, or interval narrowing, across all the variables/tree nodes of Exp. Interpreter initiates a recursive narrowing process on Exp, with the IA module handling all the interval arithmetic calculations. The narrowing process continues until the convergence criteria are satisfied; then UInterface directs the narrowed Exp back to user interface modules, which will present Exp in the forms of numbers and solution space plots.

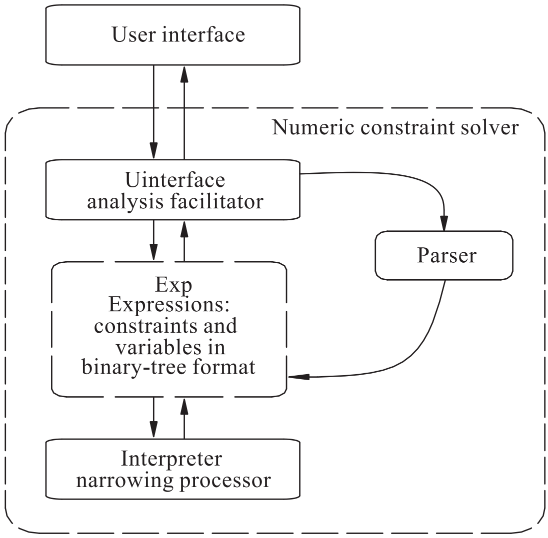

Fig. 1 Constraint solver components

Figure 2 is used to illustrate a narrowing process by QES2. First QES2 breaks down the constraint set into a binary tree form as shown in Fig. 2a. The relationships, such as (logical AND), +, –, etc. in the constraint set provide the linkages between two sub nodes. The sub nodes keep breaking down until each node is composed of a single variable. At the same time, a variable table is being assembled including all the variables in the constraint set (Fig. 2b). In QES2, every variable is defined as a real interval, representing the lower and upper bounds of the variable as a pair of real numbers.

A narrowing process takes the interval table from its initial state to the converged state, at which all variables can no longer be narrowed[5]. Using the narrowing functions provided in the interval arithmetic library, the interval of each variable is being narrowed with interval arithmetic operations. Special techniques described in the preceding sections are incorporated in the narrowing process to enforce the efficiency and stability. A narrowing pass is a recursive process starting at the root node and ending at the end nodes, with all variable intervals being updated in real time to the variable table. At the end of a narrowing pass, all variable intervals are narrowed at least once according to the constraints encountered during narrowing. In most cases, especially for complex multi-variable problems, a narrowing pass is not strong enough to stabilize the variable intervals, i.e., the intervals can be further narrowed. Thus, additional narrowing passes are requested and carried out repeatedly until a stage is reached at which all variable intervals can no longer be narrowed. This is the solution stage and all variable intervals represent the solution to the problem (Fig. 2b).

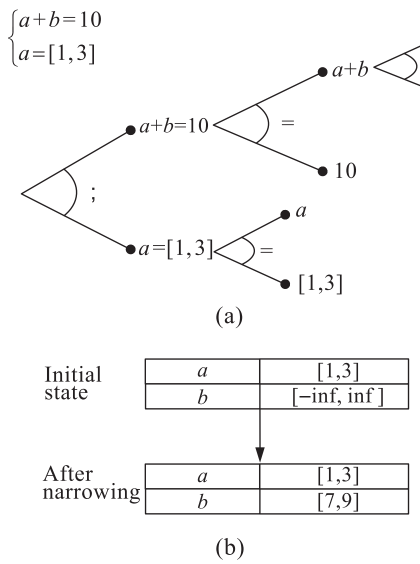

Fig. 2 Binary tree representation of a sample constraint set

2.2 Solution space plotter

The solution space plotter is tightly integrated with the main solver control module. The premise to generate a solution space plot is the availability of a successful solution set. When plotting the variable pair V_a and V_b, their solutions must be available to fill up the solution space. QES2 solves numerical equations and in-equations based on a proving process; a sub-region is valid only if its parent region is valid. Therefore, if a region of solution space is invalid in the constraint system; its sub-regions are invalid as well. The adaptive plotting method used by QES2 subdivides the solution space into quadrants. When the validity of the entire solution space of V_a and V_b is not available, the quadrant regions are invalid, resulting in a blank solution space plot.

Figure 3 is used to illustrate the process of solution space plotting. First the equations or constraints are solved by UInterface, resulting in a solution space spanning x:[−1,3], y:[0,9]. PlotAreaPanel then divides the solution space into quadrants once the validity of the entire solution space of x and y is known. It generates constraint pairs corresponding to each quadrant. Each pair of constraint sets is added to the existing solution set and stored in UInterface, which then initiates a narrowing process to prove the availability of solutions with the added constraints. If a quadrant is proven to contain no solution, it is plotted as blank; otherwise, the quadrant is further divided into smaller quadrants until the requested resolution is reached. The solution space in Fig. 3 stops at the resolution of 8×8, and shows that only quadrants containing solutions are further subdivided. The maximum resolution of a plot is physically limited by the output media, such as monitor screens and paper prints.

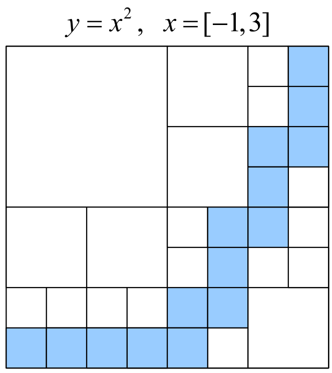

Fig. 3 Illustration of adaptive interval constraint plotting

3 Fatigue Design Example

The software framework QES2 integrates the techniques of constraint satisfaction processing, interval arithmetic, and adaptive graphing, as described in the preceding chapters. The outcome of this integration is the ability of QES2 to handle engineering problems that have complex relationships and uncertainties, and also to disclose graphically the correlations among all design variables. QES2 calls for the following requirements when modeling an engineering problem:

- The problem must be described, explicitly or implicitly, by equations that define the relationships between the design variables. The components of the equation system, or the equation system as a whole, can be linear or nonlinear, determinate or indeterminate.
- All design variables can be described numerically, in the form of a single real number, or as an interval between two real numbers.

By solving the given problem with numerical constraint reasoning, QES2 produces sound solutions, where the highest achievable numerical accuracy is limited only by the hosting software and hardware platforms. Upon completion of a successful analysis, QES2 presents the results in the following forms.

- The design variable results are expressed as intervals between two real numbers. If an interval of a design variable is too narrow to be physically meaningful, the value of the variable may be considered to be a single value; otherwise, the variable is valid inside of the interval.
- A two-dimensional graph with specified resolutions can be plotted between two design variables. Provided that the variables are valid as intervals, instead of as a single real number, the plot will be an area, or solution space, instead of a line. The two-dimensional graph further improves the representation of the output by including discontinuous solution spaces. The solution space plot is as sound as the numerical interval output, since it is produced by the same proving process.
- Multiple sets of solution spaces can be plotted on the same graph, showing the relationships between a pair of variables under different sets of constraints.

Fatigue design

Fatigue design is a critical component of any structural design involving heavy cyclical loads or movable structural parts. Some examples of structures that experience cyclical loads are bridges, material handling structures, offshore platforms, entertainment ride systems, and astronomical telescope enclosures. The aim of fatigue design is to ensure that the structure has an adequate fatigue life. Calculated fatigue life also forms the basis for efficient inspection programs during fabrication and the operational life of the structure.

To ensure that the structural engineering design and analysis, a fatigue assessment can be made by the following methods: stress-life cycle (S-N) analysis, and linear elastic fracture mechanics (LEFM) analysis. By using QES2 to carry out fatigue calculations, the following improvements can be made to help understand the studied detail:

(1) Uncertainties of various parameters can be incorporated.

(2) Solution spaces of fatigue utilization will be available for comparison of different design codes.

Two design codes are used in the following fatigue calculations: CAN/CSA-S16.1 and NORSOK. Both of these codes are based mainly on the S-N method, and are selected for illustration since they each represent a different application of the S-N method. CSA-S16.1 and NORSOK divide structural details and loading situations into a number of categories and assign different sets of parameters A, B, and m. NORSOK has more detailed categories, and also accounts for other effects in calculations of the parameters. For example, when determining stress range $\Delta\sigma$, NORSOK takes into account the effect of geometric stress concentration, or hot spot stress. When using CSA-S16.1, the studied structural detail may be approximated as detail E_1 or E, depending on the quality of the weld. This variation of detail categorization can be represented by using a corresponding range of fatigue life constants from 12.8×10^{10} to 36.1×10^{10}, resulting in a fatigue utilization range of [0.34, 0.97] calculated using QES2. In the QES2 scripts, utilf is the final design utilization, calculated by actual load cycles divided by allowed load cycles. Fymax is the maximum vertical force applied at the free end of the link arm. The variable loga considers the uncertainties in determining a proper detail category.

When using NORSOK, uncertainties exist with not only the detail classification, but also with the stress concentration factor. NORSOK's definition of detail categories puts the studied case between F_3 and W_2, with the corresponding stress concentration factor varying from 1.80 to 2.25. Such variations result in a fatigue utilization range of [1.11, 4.19]. The variations of design variables and results are summarized in Table 1.

Table 1 Parameters in fatigue calculations considering uncertainties

Formulation		
$\log N = A - m \cdot \log(B\Delta\sigma)$, where N = Fatigue life in number of cycles to failure; A, B, m = Parameters pertaining to material, detail categories, and other effects; $\Delta\sigma$ = Specified stress range		
Design variables	**CSA-S16.1**	**NORSOK**
Detail category selection	E - E_1	F_3 - W_2
A	11.107 for E_1, and 11.557 5 for E	11.261 - 11.546
B	1.0	1.80 - 2.25
m	3	3
N	2 344 044 - 6 610 938	542 774 - 2 043 379
Fatigue utilization	0.34 - 0.97	1.11 - 4.19
Design	Safe*	Fails*

To help understand the effect of such variations, the solution space using NORSOK method is plotted in Fig. 4 with the primary force F_y varying from 0 to 5 kN. The plot shows that the studied detail is by no means sufficient unless F_y is approaching zero. Using QES2's Multi-plot, the solution spaces of CSA-S16.1 and NORSOK can be combined for comparison (Fig. 4). The combined solution space shows that for the studied structural detail the outcome of the two codes intersect partially, while NORSOK is dominantly conservative with CSA-S16.1 effectively populating the lower (less conservative) bound of the solution space.

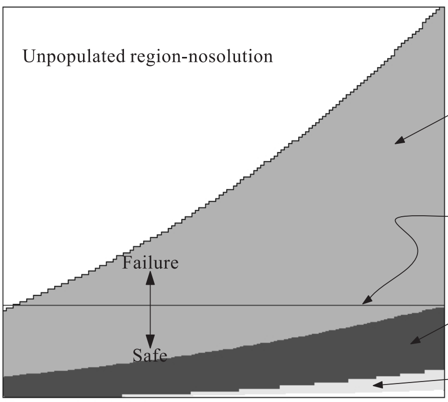

Fig. 4 Fatigue design using QES2, multiple solution space plot per NORSOK and CSA-S16.1

Aided by calculations and plots from QES2, the following conclusions can be drawn:

(1) The studied structural detail cannot be exactly described the design codes. Using conservative design assumptions, the two codes result in contrasting utilization factors.

(2) By incorporating uncertainties in the code calculations, the first conclusion still holds true with NORSOK giving the verdict of fatigue failure.

(3) Multiple code design by plotting solution space of fatigue utilization reveals that the two design codes do correlate, insofar as the code parameters in CSA-S16.1 result in a lower bound to the NORSOK results.

(4) Considering all the unknown and uncertain factors, the structural detail may be prone to fatigue failure during its service life. Proper design modification is recommended.

Although a design engineer may draw the same conclusion using plain calculations with the two codes, incorporations of uncertainties and plots of the solution space allow such a judgment to be made with additional soundness.

4 Conclusions

Qualitative reasoning techniques are effective in solving problems that can be expressed with constraints. Constraints are able to account for uncertainty in engineering design by using numerical intervals for input parameters. Being a proving process, qualitative reasoning also possesses the following characteristics that are beneficial for engineering applications:

(1) Faithfulness to the mathematical model

In conventional numerical computations, floating-point numbers in the computer substitute the true real numbers in a strictly mathematical sense. It is well documented that this can potentially result in large numerical errors. Qualitative reasoning regards numerical computations as computer-generated proofs that true real numbers are contained within a certain pair of computer floating-point numbers. This avoids any computational divergence caused by computer errors[6].

(2) Soundness of results

The reasoning operations result in intervals that contain all values that are logically possible from arithmetic operations. Consequently, the existing computer hardware can in effect prove nonexistence of solutions.

(3) No restriction on forms of formulations

Conventional numerical methods usually restrict the problem formulations to certain forms, such as linear equations or polynomials. Qualitative reasoning approaches problems by breaking down all formulations into simple logical relationships, thus in effect ignores the actual forms of problem formulations.

(4) Accommodation of uncertainties

Qualitative analysis operates on pairs of real numbers representing intervals containing all possible values. The interval expression is used in input as well as output. Engineers are able to take into account uncertainties by specifying interval bounds for the design parameters.

Fatigue design is an excellent example of an engineering problem with the aforementioned characteristics. The complex nature of metal fatigue results in uncertainties in the design process. The techniques integrated in QES2 remove the traditional concerns associated with complex formulations, thus making it possible to simultaneously use multiple design codes and accommodate uncertainty in all design parameters.

References

[1] Benhamou F, Older W J. Applying interval arithmetic to real, integer, and Boolean constraints. *Journal of Logic Programming*, 1997, **32**(1): 1-24.

[2] Gedig M H. A framework for qualitative and semi-quantitative analysis in engineering design and evaluation [Dissertation]. Canada: University of British Columbia, 1995.

[3] Hickey T. Metalevel interval arithmetic and verifiable constraint solving. *Journal of Functional and Logic Programming*, 2001, **7**: 1-25.

[4] Rossi F. Constraint (Logic) Programming: A Survey on Research and Applications. Berlin: Springer-Verlag, 2000.

[5] Zhou Y. Engineering qualitative analysis and its application on fatigue design of steel structures [Dissertation]. Canada: Univ. of British Columbia, 2003.

[6] Zhou Y, Stiemer S F. Engineering analysis with uncertainties and complexities, using reasoning approaches. *J. Comp. in Civ. Engrg.*, 2007, **21**(5): 353-362.