

SANDIA REPORT

SAND2005-3769

Unlimited Release

Printed September 2005

CONSTRUCTOR: Synthesizing Information about Uncertain Variables

Scott Ferson, Janos Hajagos, David S. Myers and W. Troy Tucker

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.osti.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND2005-3769
Unlimited Release
Printed September 2005

CONSTRUCTOR

Synthesizing Information about Uncertain Variables

Scott Ferson, Janos Hajagos, David S. Myers and W. Troy Tucker
Applied Biomathematics
Setauket, New York 11733

Abstract

Constructor is software for the Microsoft Windows microcomputer environment that facilitates the collation of empirical information and expert judgment for the specification of probability distributions, probability boxes, random sets or Dempster-Shafer structures from data, qualitative shape information, constraints on moments, order statistics, densities, and coverage probabilities about uncertain unidimensional quantities. These quantities may be real-valued, integer-valued or logical values.

Table of Contents

Acknowledgments.....	5
Hardware and software requirements	5
Getting started.....	5
Technical support.....	5
Citing the software.....	5
1. Introduction.....	6
1.1 Dempster-Shafer structures and random sets.....	6
1.2 Probability boxes (p-boxes).....	7
1.3 Using this document	8
2. Summarizing information in probabilistic analyses	9
2.1 Program tutorial: specifying input	10
2.1.1 Name and units	11
2.1.2 Parameter bounds.....	14
2.1.3 Percentile ranges	21
2.1.4 Graphical limits on the distribution	23
2.1.5 Shape constraints on the distribution	25
2.1.6 Density bounds.....	30
2.1.7 Coverage probabilities	33
2.1.8 Data	36
2.2 Exploratory mode.....	41
2.3 Justifications	43
2.4 Input and editing	50
2.4.1 Editing and formatting text entries	50
2.4.2 Entering and editing data in spreadsheets	52
2.4.3 Clearing and resetting entries.....	53
2.5 Units.....	53
2.6 Programmatic limits and specifications	54
3. Output.....	55
3.1 Synthesizing constraints to select distributions	55
3.2 Program tutorial: getting output.....	56
3.2.1 Pinching	59
3.3 Maximum entropy.....	63
3.4 Wrangling the graphs.....	65
4. Handling imprecise data.....	66
4.1 Stochastic mixture.....	66
4.2 Solana and Lind's sample rule.....	68
4.3 Relaxed sample rule.....	69
4.4 Kolmogorov-Smirnov	71
4.5 Saw-Yang-Mo.....	72
4.6 Rigorous bounds from sample data.....	75
5. Options and program settings.....	78
6. Automatic input cascading.....	81
7. Frequently asked questions.....	83
8. References.....	86

Acknowledgments

The software, including the executable programs and this documentation, were created under funding from Sandia National Laboratories (Contract Number 19094), as a part of the Epistemic Uncertainty Project directed by William Oberkampf, and under funding from the National Institutes of Health Small Business Innovation Program (grant 5R44ES010511-03). William Root provided consultation on Windows programming. Many helpful suggestions were made by William Oberkampf, Jon Helton, Lev Ginzburg, Cliff Joslyn, Kari Sentz, Alyson Wilson, Jim Hall, Marty Pilch, Igor Kozine, and Vladik Kreinovich. Laura Swiler and Richard Recharad read the documentation and suggested several improvements. The opinions expressed herein and embodied in the program are those of the authors and do not necessarily reflect the positions of Sandia National Laboratories or the National Institutes of Health.

Hardware and software requirements

Constructor should run on most Windows 95, 98, 2000, NT and XP installations. It may also run on emulations of Windows under other environments.

Getting started

Download the file SETUP.EXE from <http://www.ramas.com/constructor.htm#download>. Run this file by double clicking on it. It is an automated installation program. It will ask which folder you would like Constructor to be placed in and copy the files you'll need. For your convenience, a shortcut to Constructor can be placed on your desktop. You can move this shortcut to a folder or delete it altogether without affecting Constructor. You can invoke Constructor by double-clicking on the shortcut, or selecting Constructor/Constructor.exe from the Start/Program menu. A two-part program tutorial starts on pages 10 and 56. Answers to frequently asked are given on page 83.

Technical support

Requests for assistance, bug reports, suggestions or comments about the software are all welcome and can be sent to Troy Tucker (troy@ramas.com, 1-631-751-4350, fax -3435). We cannot guarantee to fix your problem, but most issues are fairly simple to resolve. For answers to frequently asked questions, links to more information about probability bounds analysis and Dempster-Shafer theory, software updates, and amendments to this document, consult the website <http://www.ramas.com/constructor.htm>. The Epistemic Uncertainty Project's website is www.sandia.gov/epistemic/.

Citing the software

The full citation for the software is currently

Ferson, S., J. Hajagos, D.S. Myers, and W.T. Tucker (2005). *Constructor: Synthesizing Information about Uncertain Variables* (<http://www.ramas.com/constructor.htm>). Applied Biomathematics, Setauket, New York.

We would appreciate learning of any applications of the software or papers or reports that may cite it. Please send information about such applications or citations to Troy Tucker (troy@ramas.com, 1-631-751-4350, fax -3435).

1. Introduction

Constructor is software for the Microsoft Windows microcomputer environment that helps users to collect empirical information and expert judgment relevant for specifying probability distributions, probability boxes, random sets or Dempster-Shafer structures on the real line. It takes as input user-supplied sample data, qualitative information about distributional shape, theoretical or inferred constraints on moments, order statistics and probabilistic coverage statements. It synthesizes these disparate forms of information and expresses the accumulated knowledge as distributions or bounds on distributions, either of which can be used in subsequent calculations outside the program. The input can be graphical or numerical. It can be specified as precise numbers or interval ranges to represent epistemic uncertainty. Constructor makes the methods for representing sparse information as probability boxes and Dempster-Shafer structures that were reviewed in Ferson et al. (2003) accessible in interactive software for desktop computers.

In cases where the supplied information is insufficient to determine a precise probability distribution, a user can employ an additional criterion to select a privileged distribution from among a family of possible distributions. The software supports six different criteria for this purpose, including maximum entropy, maximum dispersion, spanning and variously conservative criteria. Alternatively, the entire family of possible distributions can be summarized as a probability box (p-box) or a random set or Dempster-Shafer structure on the real line.

The primary use of Constructor is to determine what can legitimately be inferred derive from sparse and sometimes disparate pieces of information about an uncertain quantity. The software is not designed as a tool for expert elicitation, although it may be useful in an elicitation process.

1.1 Dempster-Shafer structures and random sets

This document assumes the reader is already familiar with the basic ideas of Dempster-Shafer evidence theory and of Dempster-Shafer structures on the real line (Helton et al. 2004; Yager 1986; Ferson et al. 2003). See Halpern (2003), Ayyub and McCuen (1997), or Klir and Yuan (1995) for an introduction to the theory, and consult Dempster (1967) and Shafer (1976) for details. A parallel development under random sets theory (Matheron 1975; Robbins 1944; 1945) is possible, leading to a random set on the real line. Several applications of these parallel theories have been made in engineering contexts (e.g., Castleton and Luo 1992; Luo and Castleton 1997; Tonon et al. 1999; 2000a,b; Oberkampf and Helton 2002; Oberkampf et al. 2001; Tonon 2004; Helton et al. 2004).

Ferson et al. (2003) surveys Dempster-Shafer structures or random sets as “uncertain numbers” and reviews the methods employed in Constructor to specify them from imprecise empirical and expert information. In Constructor, Dempster-Shafer structures and random sets are assumed to be composed of finitely many closed intervals from the real line with associated nonnegative masses which sum to unity. Figure 1 depicts an example of such a structure. The values within each interval are empirically indistinguishable from other values within the interval because of measurement uncertainty or similar cause. If the widths of all the intervals decrease to zero, then the Dempster-Shafer structure converges to a discrete probability distribution.

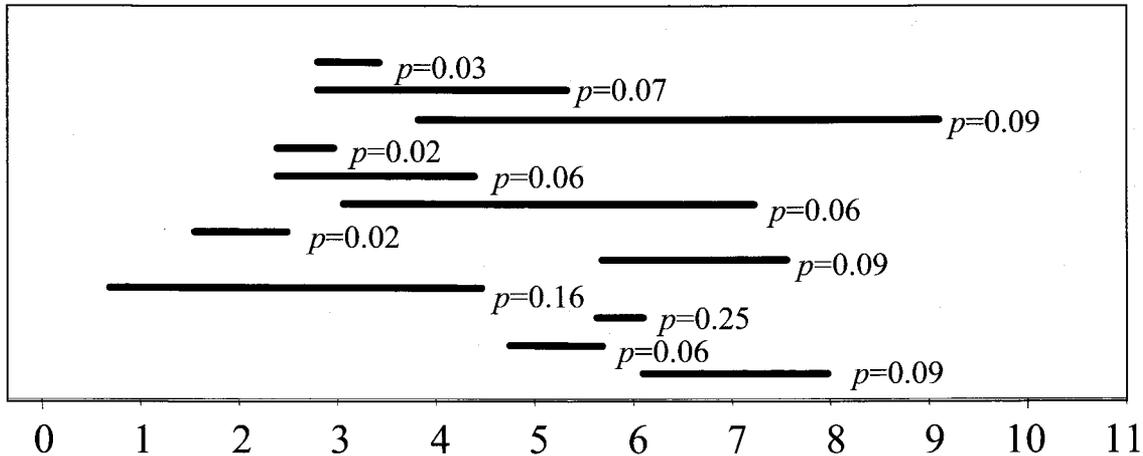


Figure 1. Example of a Dempster-Shafer structure. Intervals of the real line (abscissa) are depicted along with their masses which sum to one. The vertical locations of intervals are arbitrary.

Yager (1986) showed how Dempster-Shafer structures could be combined in arithmetic operations that generalize convolutions of probability distributions. Such calculations can be used in risk analysis and other applications of probabilistic uncertainty assessment when the available empirical information is not sufficient to warrant the use of precise probability distributions. Berleant and Zhang (2004) and Ferson (2002) offered software implementations.

1.2 Probability boxes (p-boxes)

This document also assumes the reader is already familiar with the basic ideas of probability bounds analysis and of probability boxes (p-boxes). Consult Tucker and Ferson (2003) or Ferson et al. (2003) for an introduction. Ayyub (2002) provides a synoptic overview of the ideas. Several applications of the theory have been made in risk analysis (e.g., Kriegler and Held 2004; Bernat et al. 2004; Ferson and Hajagos 2004; EPA 2003a,b; Ferson and Tucker 2003; EPA 2002; Regan et al. 2002a,b; Kaas et al. 2000; Goovaerts et al. 2000). See also Berleant and Zhang (2004) and Fetz and Oberguggenberger (2004) for related ideas.

Consult Ferson et al. (2003) for a review of p-boxes as uncertain numbers and a review of the methods employed in Constructor to specify them from imprecise empirical and expert information. Figure 2 depicts three examples of p-boxes. The first is a degenerate p-box because it is a precise probability distribution shown as a cumulative distribution function. The third is also a degenerate p-box because it is an interval. The middle p-box is a structure that is neither a distribution function nor an interval but has characteristics of both. A p-box can be identified with a class of distribution functions, each of whose cumulative distribution functions resides entirely within the bounds defining the p-box. Any Dempster-Shafer structure or random set on the real line can be depicted as a p-box where the upper bound is the cumulative plausibility function and the lower bound is the cumulative belief function. P-boxes can be used in quantitative risk analyses and other applications of probabilistic uncertainty modeling, and software tools exist for this purpose (Ferson 2002; Berleant and Zhang 2004).

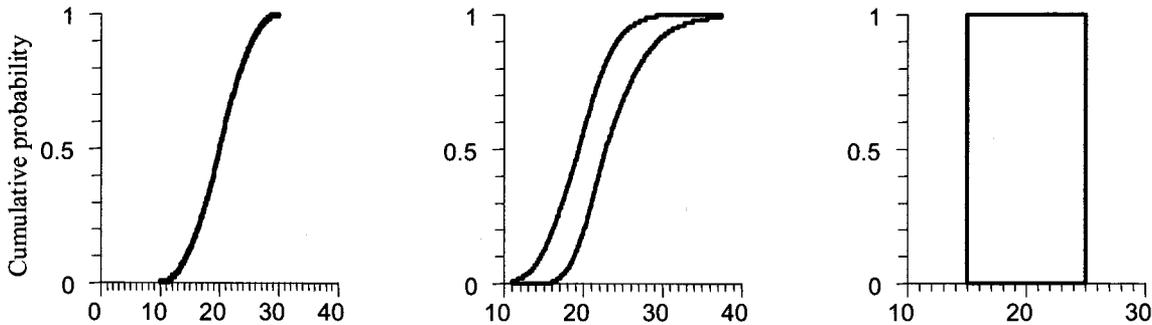


Figure 2. Three examples of probability boxes (p-boxes). For each graph, the ordinate is cumulative probability and the abscissa is the value of the (imprecisely characterized) random quantity.

1.3 Using this document

This document serves as both the user manual and reference guide for the Constructor software. Although it presumes familiarity with the methods reviewed in Ferson et al. (2003), it is designed to be reasonably self-contained. It attempts to provide complete documentation for all the capabilities, program features and options available within the software. Consequently, this document is much longer and perhaps more daunting than a beginning user needs to become familiar with and use the software. The associated website <http://www.ramas.com/constructor.htm> (which is available from within the Constructor software by selecting Help/Website from the main menu) is a highly abbreviated version of this document, and it may be useful if you want a fairly quick introduction to what Constructor does and how you can use it. You can consult this document if you need details about program use and more thorough descriptions of program features, if you are not already easily familiar with Windows-based software, or if you just prefer reading printed material.

This document includes tutorials on specifying input (page 10) and getting output (page 56) which walk through many of the features of Constructor and exemplify the uses you might make of the software. The first tutorial, spanning over thirty pages, seems rather long, but because almost every page has a large figure, the actual text to read is only about twelve and a half pages. It can easily be digested in under an hour. Although this document can usefully be read like a book from beginning to end, for many users it might be most convenient to read through the two tutorials first and then treat the other material as a reference which you consult only when a specific question arises.

Section 2 describes the input for Constructor. This includes the first program tutorial and also descriptions of exploratory mode, the justification system, and details about editing inputs. Section 3 describes Constructor's output, including the second tutorial, a discussion about the maximum entropy criterion, and points on how to format graphical output. Section 4 is a discussion about how sample data can be used to characterize uncertain numbers. Section 5 describes the various options and program settings under user control. Section 6 summarizes Constructor's scheme for automatically integrating information across the various input pages. Section 7 gives answers to some frequently asked questions about the software.

2. Summarizing information in probabilistic analyses

Whenever inputs for probabilistic analyses must be selected, or output from analyses need to be summarized, we have to specify an uncertain number* from imperfect information. Virtually all of the relevant technical literature concerns selecting probability distributions, but the broad approaches and themes of this literature apply to the general situation of choosing uncertain numbers. These approaches include use of defaults, deference to expert opinion, and inference from constraints. When the available empirical information is limited and technical understanding is incomplete, these ways can often lead to rather different selections, so it is incumbent on an analyst to choose the method conscientiously. This section will briefly introduce these ways as they have been used to select probability distributions, and then provide an extended software tutorial that introduces the features of Constructor by which users specify inputs to summarize probabilistic information.

In many disciplines, the most common method for selecting probability distributions is to rely on defaults. Default distributions are simply those which are traditionally used for some purpose. In some cases, their use was originally well motivated by available empirical information and physics-based considerations. In other cases, distributions which were originally proffered as mere hypotheses became, by repeated invocation, the standard model to use. In this sense, like so many other elements in modern life, they are famous simply for being famous. The advantage for analysts in using default distributions is two-fold. First, they are easy to choose because they come out of a book, or engineering literature, and there is no real selection process. Second, they generate very little controversy under review. Everyone realizes that their purpose is to be a placeholder for a distribution whose true nature is unknown.

The disadvantages of default distributions, on the other hand, are also two-fold. The first is that they are not well motivated and are probably wrong. This fact may be obscured by their incorporation into a complex analysis, the results of which could also be wrong by the fact of their depending on the defaults used as inputs. The second disadvantage is that selecting default distributions does not allow the analyst to make use of any information that may be available. The use of default distributions is still an important approach, but the purpose of Constructor is to make alternative approaches more accessible.

Another very common means for selecting input distributions is to allow sample data to specify an empirical distribution function (EDF). This idea is reviewed in the section "4.1 Stochastic mixture" on page 66. The advantage is that the choice can be automatic so that little thinking is required. The disadvantage is that there is often little empirical data on which to base the distribution. In most cases, EDFs are only samples from the distribution of interest. Even if the samples are unbiased and precise, they contain sampling uncertainty. The fewer the samples, the greater the sampling uncertainty. When the reliability and representativeness of the available data is questionable, using an EDF may not be appropriate.

*Uncertain numbers include probability distributions, Dempster-Shafer structures or random sets over the real numbers, p-boxes and intervals.

In many cases, it is possible to extrapolate a distribution from surrogate data. For instance, one may need a distribution for failure temperatures for a new kind of switch and try to estimate it by reference to an observed distribution of failure temperatures for a different but similar kind of switch for which there is abundant data. The drawback of this approach is that it generally depends on professional judgment and sometimes a considerable investment of the analysts' modeling effort to produce the best possible result, which depends finally on professional judgments which may turn out to be erroneous.

In some fields, elicitation from experts is the predominant means by which distributions are selected. Instead of using the analyst's own judgment, he or she makes use of the knowledge of others who have been recognized as experts for one reason or another. This process can be cumbersome and expensive, and it can yield controversial results when experts disagree, as they often do when empirical information is sparse (see Johnson et al. 1982; Hammond 1996). Many performance studies have shown that experts tend to underestimate their own uncertainty, and the result may be distributions that show less variability than really exists (Henrion and Fischhoff 1986; Morgan and Henrion 1990; Plous 1993).

The maximum entropy criterion (Jaynes 2003; Lee and Wright 1994) is also widely used, especially among Bayesians, for selecting probability distributions when information is very sparse. The use of this criterion is discussed in the section "3.3 Maximum entropy" beginning on page 63. Its main advantage is that it selects a distribution that is least biased with respect to what is unknown, but its central disadvantage is that it yields inconsistent results under changes of the underlying scale.

The section below is a program tutorial that walks through the various input pages on which you tell Constructor what you know about the uncertain quantity. It will give you an overall understanding of the kinds of information that can be employed. After the program tutorial, there is a short review of various ways to handling imprecise sample data, and a synopsis of the keystrokes and shortcuts for editing and formatting the entries you make in Constructor.

2.1 Program tutorial: specifying input

Constructor is designed so that you can enter several kinds of information about an uncertain quantity. You can specify bounds on the parameters of the distribution if you know them, or bounds on percentiles, or bounds on probability densities. You can also specify qualitative features about the shape of the distribution. You can also prescribe coverage probabilities for interval ranges. You can specify any empirical sample data that may be available, and you can even graphically constrain the distribution function. You probably won't have each of these kinds of information about a single quantity, and indeed you don't need to have any particular information to use Constructor. You can specify whatever information or judgments you have on eight input pages.

In the several sections below, we will walk through Constructor's eight input pages. We won't cover the pages in the order they are displayed, but will instead consider them in an order that is slightly more pedagogically useful. This walk-through assumes that you have already downloaded and installed the software on your microcomputer. (See the section "Getting started" on page 5 if you haven't done this yet.) Invoke Constructor by clicking on the Start menu (in the lower, left-hand corner of your computer screen)

and selecting All Programs/Constructor/constructor.exe from the popup menus. Alternatively, you can invoke Constructor by double-clicking on the Constructor icon that appeared on your desktop after you installed the software.

After you invoke Constructor you may see a briefly displayed banner page, after which the display will look like Figure 3. The main menu (“File Edit Input Justification Bounding Help”) appears at the top of the display. Below the main menu is a graph (which is initially blank), and below the graph are the tabs by which you can access the eight input pages. The first of these is the Name and units page, which consists of six (initially blank) input fields. At the bottom of the display are some shortcut buttons.

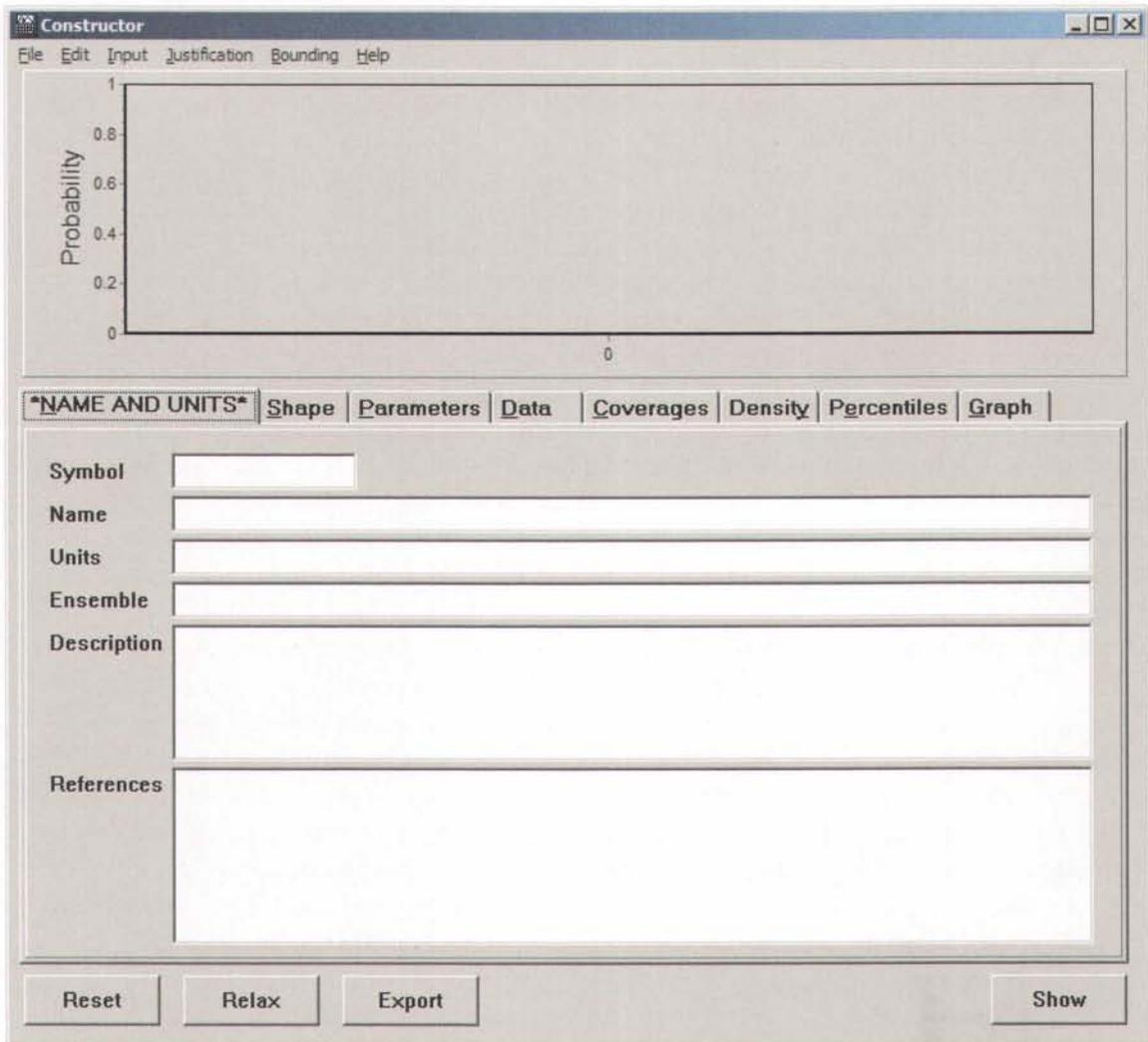


Figure 3.

2.1.1 Name and units

When you first invoke the Constructor program, and whenever you click on the “Name and units” tab, you will see a display like Figure 3 above. The white fields in this display are where you should start to specify the uncertain number you are using Constructor to characterize. If the uncertain number has a symbol, you can enter it on the first field.

(The field is short as a suggestion that symbol names should generally be short.) You can spell out the name of the uncertain number without abbreviation in the second field.

In the third field you can specify the default units for the uncertain number. If you don't specify units for the particular values you enter on other input fields, Constructor will assume that you intend them to be in units that conform with the units you specify here. Also, the output produced by Constructor will be expressed in terms of these units. Note that, unlike most programs, Constructor uses the units you specify in calculations and does not treat them as mere labeling. You can specify exponents for units in curly braces or with the words "square" or "cubic". The words "per" and "over" are also understood. For example, the units "square meters", " m^2 ", "meter²" and even "m meters" are all the same. Likewise, the expressions " $kg\ m\ s^{-2}$ ", "kilogram meter per square second", " $kg\ s^{-2}\ m$ ", " $kg\ m\ over\ s^2$ ", "kilogram per second per second meter", "newton", "N" are all the same.

In the fourth field, you should enter the statistical ensemble for the uncertain number. The ensemble is the statistical population associated with the variation that the uncertain number is intended to characterize. The ensemble is also known among probabilists as the reference class. If the uncertain number you're modeling is a characterization of a random variable such as, say, the set of temperatures that a component will experience during its operating life, then the ensemble will be the distribution of temperature values that could be (or have been) measured during operation. If the uncertain number characterizes, for instance, the variation in component reliability, the ensemble might be the set of durations of operating lifetimes that similar components will exhibit. If the uncertain number is a characterization of a *constant* rather than a random variable, the ensemble might be various measurements of its value. Although the constant presumably has a fixed magnitude, different measurements of it will vary slightly and it is the distribution of these varying measurements that is (usually) considered to be the statistical ensemble when the constant is characterized by a probability distribution or other uncertain number.

One of the most serious problems in risk analyses and other applications of probabilistic uncertainty modeling is confusion about the ensembles of the variables involved in the analysis. Sometimes analysts want to consider a problem abstractly and not specify such information, but any serious model of uncertainty must be completely clear about these details. Glen Suter (*voce*) has argued that many models in risk analysis are nonsensical because they have neglected to specify information as basic as whether the variation is spatial or temporal. Gigerenzer (2002) offered a memorable example of how the ensemble can make a big difference in how a statistical quantity is interpreted. Medical doctors advise patients having prostate surgery that the rate of sexual dysfunction after such surgery is about 50%. Some patients understand this to mean that, on average, one out of every two of their attempts to have sex will be unsuccessful. In fact, what it means is that one out of every two *patients* will be totally impotent. The ensemble that the failure rate describes consists of patients rather than sexual bouts. Obviously, there is a profound difference for a patient between a substantial risk of being completely dysfunctional and having a substantial diminution of sexual activity. The difference will be misunderstood if the ensemble is not clearly described.

Suppose that the quantity under consideration is a failure rate of a component, which is, say, roughly 3%. Even though we may have a complete numerical characterization of

the quantity, we cannot use the characterization in calculations unless we also know what statistical ensemble it represents. What component does it refer to? What are the units of the failures? Per demand? Per lifetime? Per component? Something else? Does the variation around 3% represent fluctuations among components? Through time? What stretch of time? Duration of the mission or something else? Across different assemblies? Which population of assemblies? To be clear about specifying the ensemble, it will often be important to specify the individuals that make up the distribution and explicitly state their measurement units. It may also help to indicate the *cardinality* of the ensemble. Knowing how many individuals comprise the ensemble or reference class allows humans to make better mental calculations about probabilities (Gigerenzer 2002).

In the fifth and sixth input fields on the page, you can enter a description of the uncertain number and relevant references respectively. The description can include any details that may be important to understanding the empirical basis on which the characterization of the uncertain number is founded. The references can include formal literature citations, details of personal communications, or general notes about where the underlying information came from.

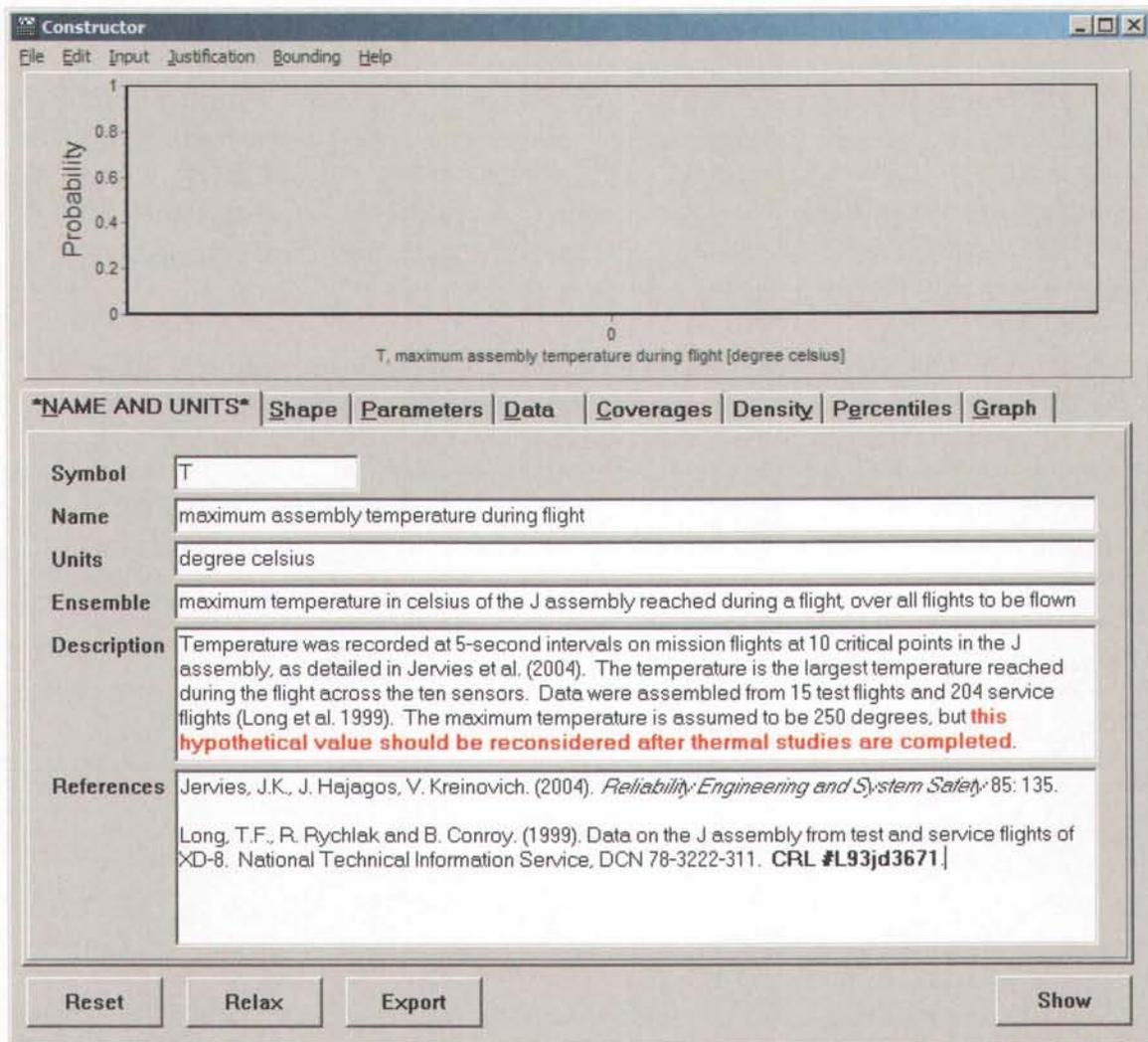


Figure 4.

The display in Figure 4 is an illustration of what the page might look like when it has been filled out. Other than the units, none of the information you give on this page influences the quantitative characterization of the uncertain number produced by Constructor. However, it can nevertheless be very important for the interpretation and documentation of the results and, as a general rule, you should try to be conscientious about entering this supporting information. The ensemble, for instance, is often neglectfully described or omitted altogether, but this can be a critical piece of information.

The text you enter in these input fields is not limited by the size of the displayed boxes. The fields can contain arbitrarily long strings. If you enter more than can be seen given the width of the display, you may wish to resize the display by clicking the full-screen icon (immediately to the left of the close-program icon that appears as an \times in a box in the upper, right-hand corner of Constructor's display window).

The two multi-line input boxes labeled "Description" and "References" are special in that you can introduce formatting for the text you enter. You can use color, italic, boldfacing, underlining, and various fonts for emphasis. You toggle on or off italic, boldfacing and underlining by pressing the Control-I, Control-B, and Control-U keys respectively. You can change the color or other characteristics of the font by invoking the font dialog by pressing Control-F. Long lines will automatically wrap and the text will scroll as needed. You can enter as much information as you like in these fields.

2.1.2 Parameter bounds

We'll skip over the Shape page for now and consider the Parameters input page. Clicking on the Parameters tab brings up the display in Figure 5.

Information about parameters can be specified here to characterize the uncertain number. An input can be a real number, an interval, or left blank to represent ignorance about that parameter. *If a parameter is unknown, just leave it blank.* Or, if you know only broad constraints on it, such as "it must be positive" or "it's got to be less than 100", then you can represent what you are sure about. If you only know one bound, you can specify it with an inequality sign. For instance, if the parameter is surely positive, you might enter " >0 ". Similarly, if it's surely less than 100, enter " <100 " to represent this fact. If the parameter has both constraints, you can simply enter the interval "[0,100]". Constructor understands the conventional computer format for scientific notation, in which, for example, $2.3e-3$ denotes $2.3 \times 10^{-3} = 0.0023$.

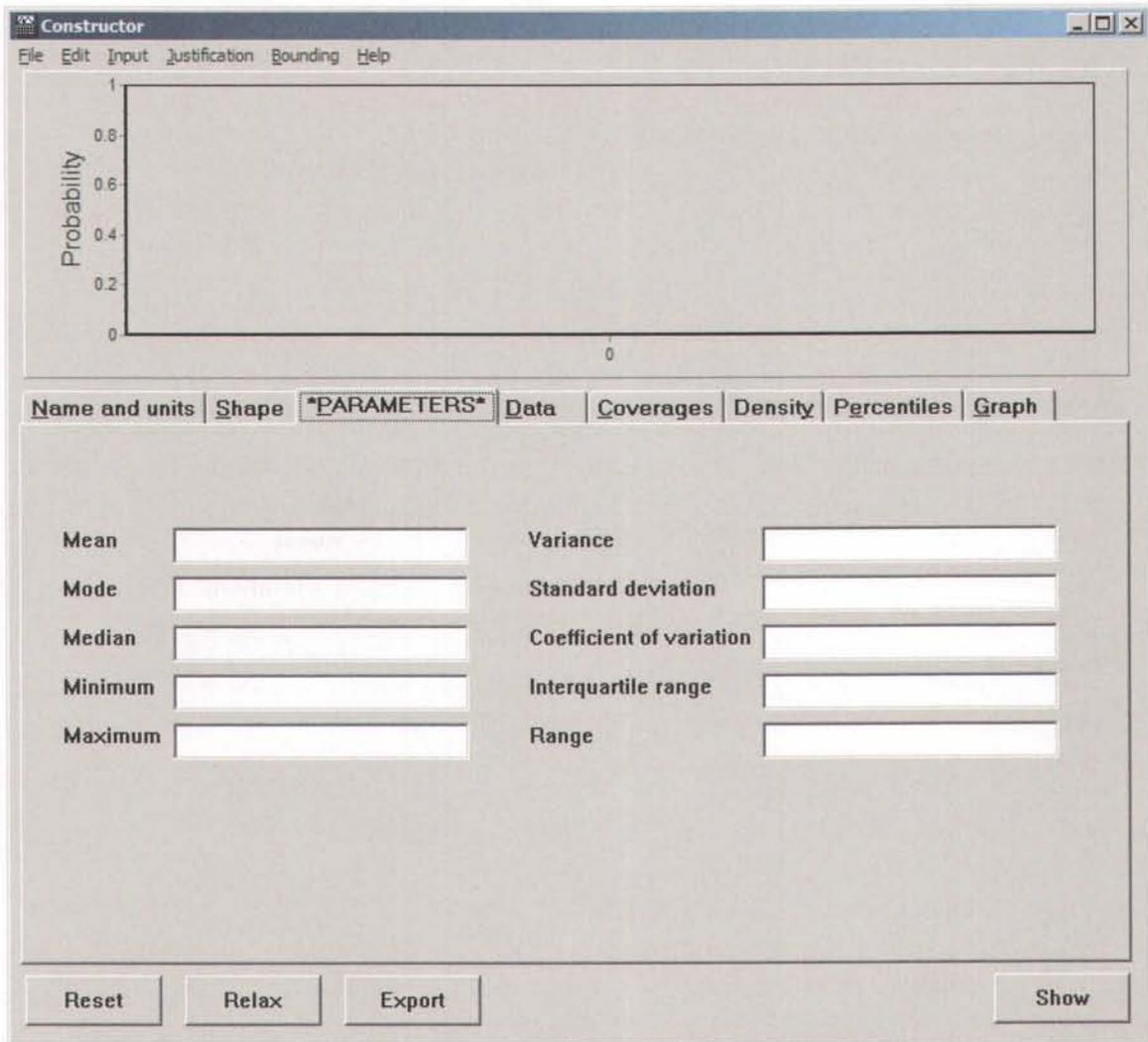


Figure 5.

Any entries that you make will automatically cascade to update other entries and the graph. Suppose, for instance, that we know the mean to be some value bigger than 4 and smaller than 6. To indicate this, click on the white mean field and type “[4, 6]” (without the quotes). Then press the Enter key to make Constructor interpret what you typed. The display should then look like Figure 6.

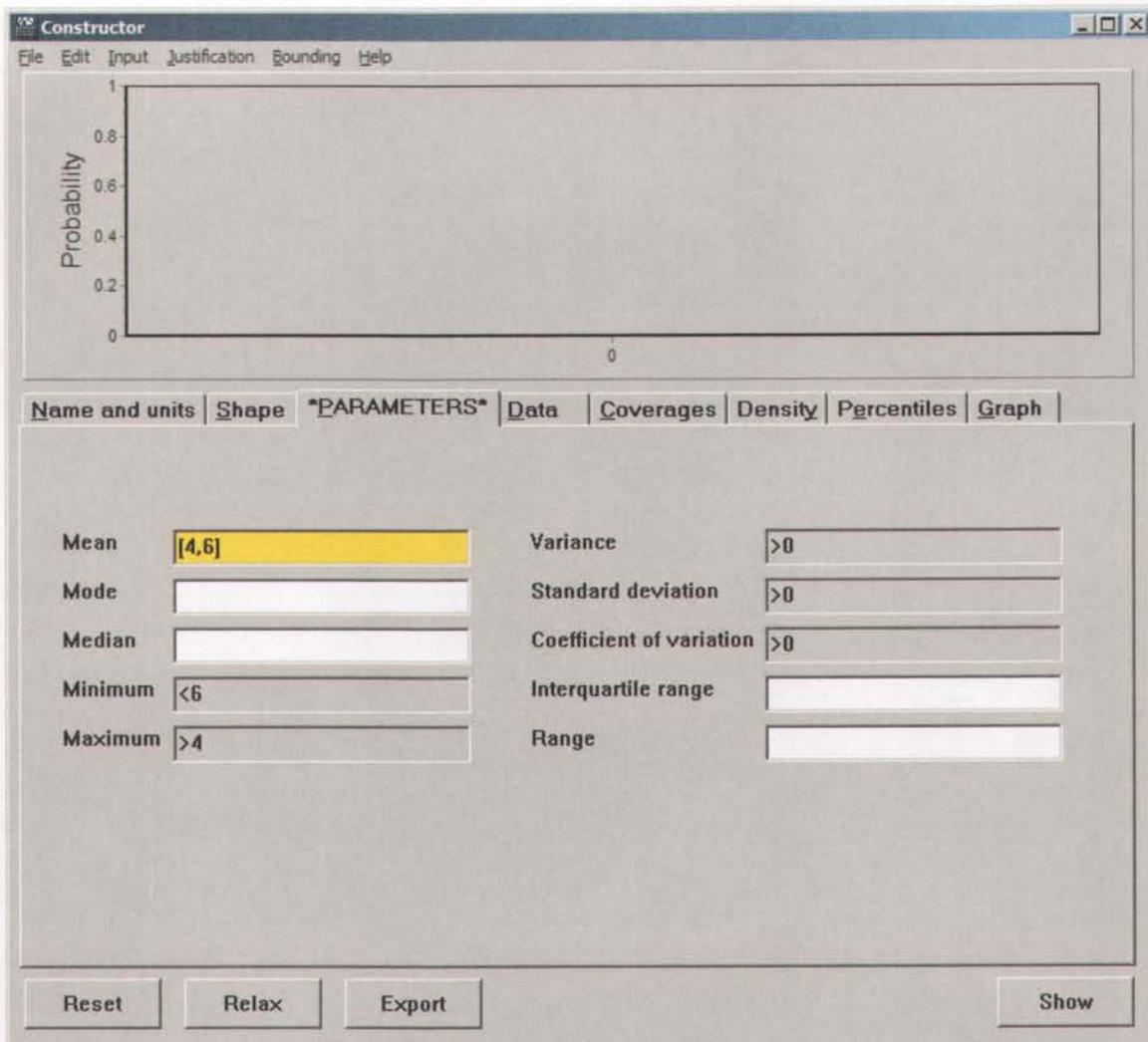


Figure 6.

Notice that Constructor has inferred some things about some of the other parameters. Telling it that the mean is between 4 and 6 lets the program infer that the minimum has to be less than 6. (Do you see why it did *not* infer that it had to be less than 4? Doing so would have been a grave mistake. It would have wrongly excluded a distribution with, say, mean 5.5 and zero variance, which is entirely consistent with the bounds on the mean.) It likewise inferred that the maximum has to be greater than 4. It also inferred that the variance, standard deviation and coefficient of variation are positive, but the first two inferences it could have made without any inputs from you.

Now press the Tab key a few times so that the cursor is in the minimum field and <6 is highlighted. If you then press the 0 (zero) key, the numeral will replace the <6. If you then press the Tab key again, several things will happen. The minimum field color will change to yellow. The inferred ranges for the other parameters in the gray fields will tighten substantially. And a graph will appear in the upper part of the display. The display will then look like Figure 7.

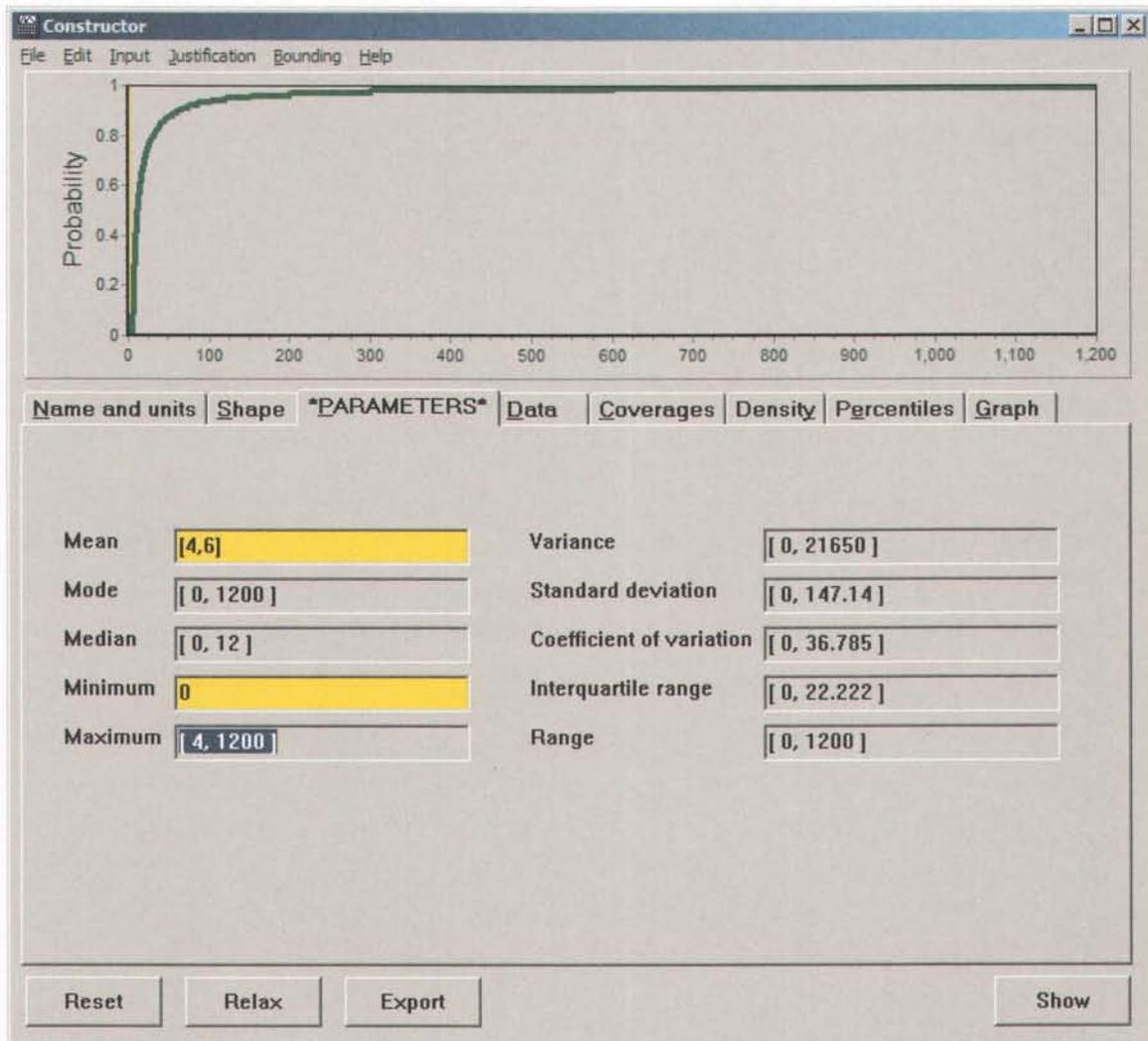


Figure 7.

It may be hard to discern at this point, but the graph shows the right (lower) bound on the cumulative distribution function (CDF) for the uncertain number based on the information you have so far characterized. Just knowing the variable is positive and its mean is no larger than 6 tells us that its CDF cannot be any further to the right that the green bound depicted in the graph. This graph is a result, as explained in Ferson et al. (2003), of the classical Markov inequality which gives constraints on the all percentiles of the distribution. The median and the interquartile range are just special cases.

Constructor also proffers loose bounds on the mode, variance, standard deviation, coefficient of variation, and range. These bounds all arise from an assumption that the maximum is no larger than 1200. This limit is not absolute. In fact, the theoretical limit of the maximum is actually infinite. The value of 1200 comes from the presumption made by Constructor that the upper bound on the 99.99th percentile (rigorously obtained from the Markov inequality) is a practical guess for the maximum of a real-world distribution. (This presumption is not always appropriate. If you don't want Constructor to make this presumption, you can prevent it from doing so by entering both the minimum and maximum before you enter the mean.)

If you now specify the maximum as, say, 20 and press the Enter key, the display will look like Figure 8 below.

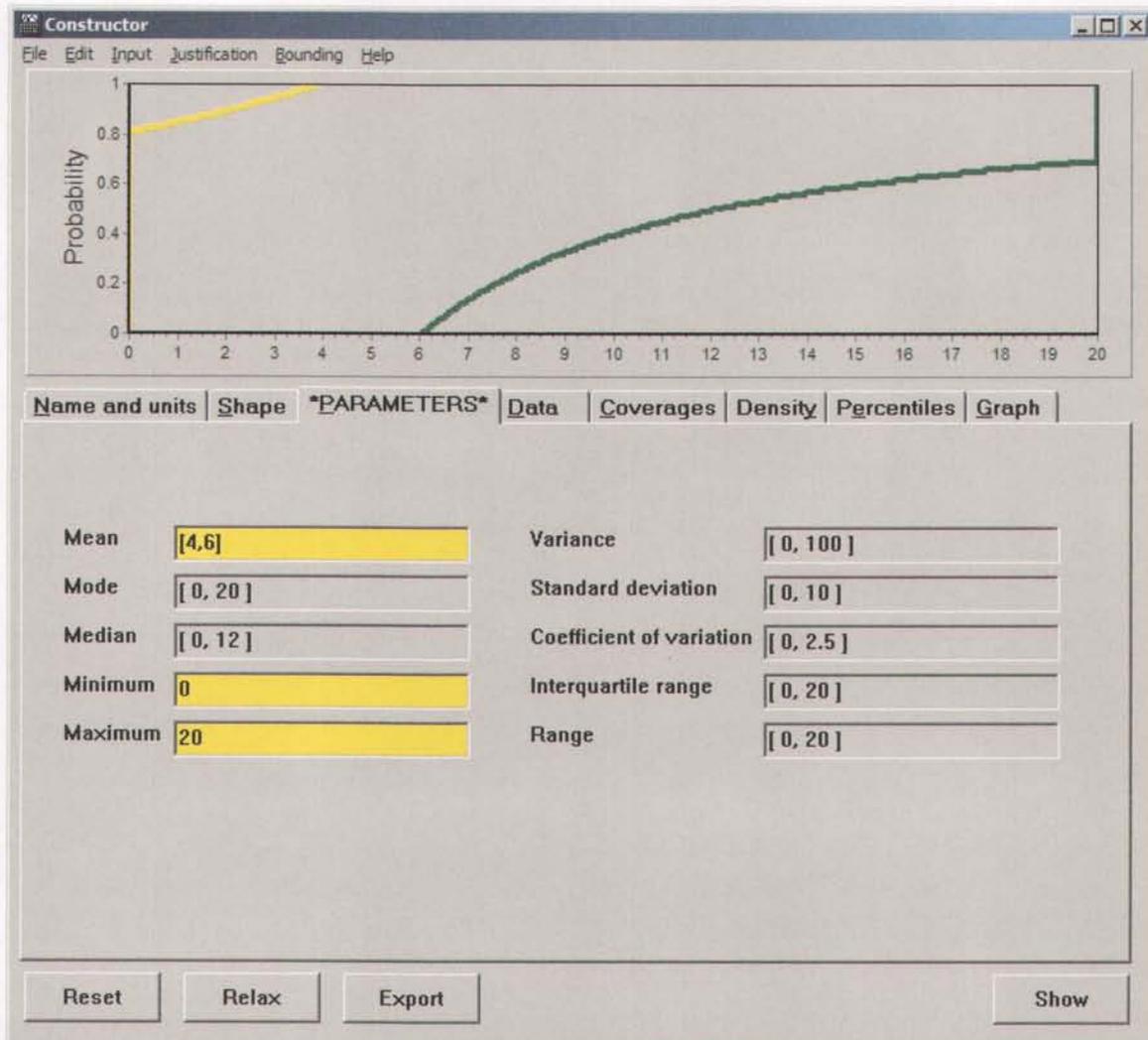


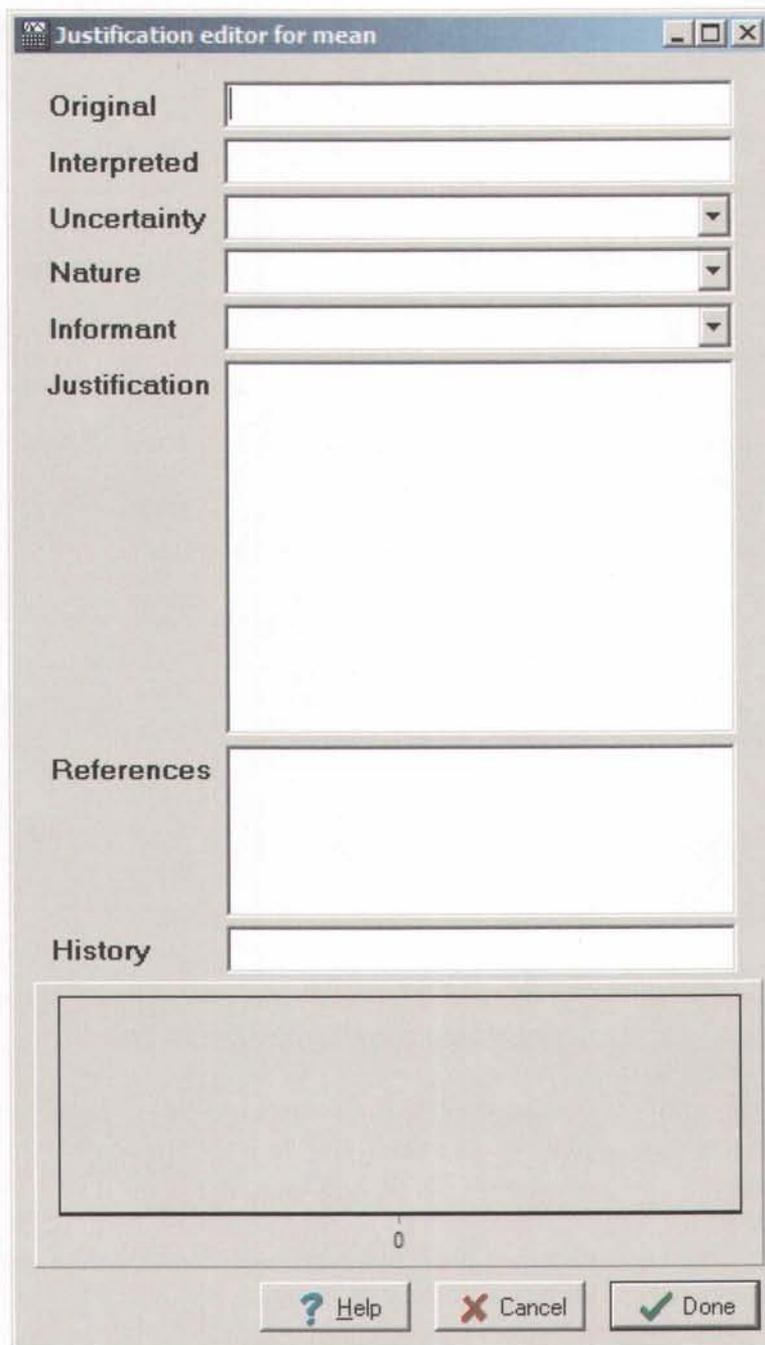
Figure 8.

Although we have only specified the minimum, maximum and mean, Constructor has inferred rather tight bounds on the remaining parameters. The bounds on the CDF at the top of the display are now easier to discern. The left (upper) bound is shown in yellow, and the right (lower) bound is shown in green. Together, these bounds are called the probability box, or p-box, for the uncertain number. The fields in which you directly specified values are colored yellow. Those fields that were inferred from the three values are colored gray.

In principle, you can use your own judgment as the analyst to determine the values to enter for the various parameters. Alternatively, you could appeal to expert opinion to specify the parameters. You may even have estimates based on empirical data. In some cases, you may have access to the original data. In other cases, the estimates may be available only from the scientific or engineering literature or other archived sources. You

may want to enter confidence intervals for the parameters rather than their point estimates to capture the sampling uncertainty associated with limited sample sizes.

In practice, you should be sure to record where the specifications for the parameters come from (your judgment, experts, or data) in the justification window for each parameter you enter. To do this, just right-click on a parameter's yellow entry field to invoke the justification dialog where you can spell out why you made the entry you did. For example, when you right-click on the mean field, you'll see a justification dialog like that shown in Figure 9.



The image shows a dialog box titled "Justification editor for mean". It contains several input fields and a large text area. The fields are labeled "Original", "Interpreted", "Uncertainty", "Nature", and "Informant". The "Justification" field is a large text area. Below it are "References" and "History" fields. At the bottom, there are three buttons: "Help", "Cancel", and "Done".

Original	<input type="text"/>
Interpreted	<input type="text"/>
Uncertainty	<input type="text"/>
Nature	<input type="text"/>
Informant	<input type="text"/>
Justification	<input type="text"/>
References	<input type="text"/>
History	<input type="text"/>

0

? Help X Cancel ✓ Done

Figure 9.

We'll review the various components of this dialog on page 41, but, for now, just type the phrase "because I said so" in the large white box labeled "Justification" and click on the Done button to close the dialog. After it closes, the display should look like Figure 10.

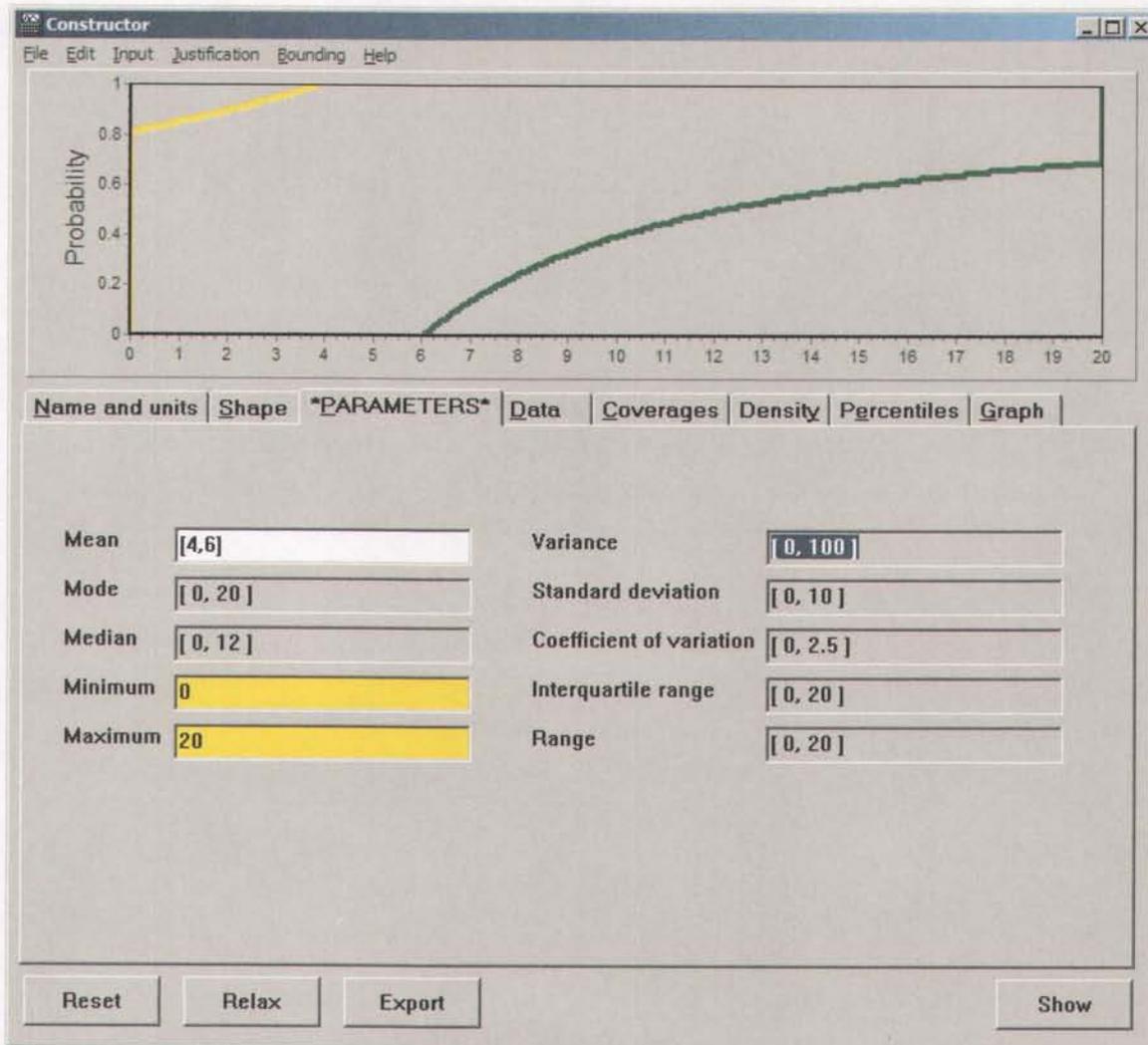


Figure 10.

The mean field, which had been colored yellow after you entered the interval, has now become white again. The white color indicates that the field's input is complete in the sense that it has been justified.

When you position the mouse over the white mean field and hold it there for a while, a little hint will appear that echoes the justification you just entered on the justification dialog. In contrast, if you hold the mouse over a yellow field, you'll be reminded that the input still needs justification. Values appearing in gray fields are inferences from other entries you've made and don't need explicit justifications. Instead, they have histories which trace the inference or inferences from which they were derived. Holding the mouse over a gray box for a while will bring up a hint that shows this history. By default, the hints persist for about 10 seconds and then disappear. If you want them to

persist longer or disappear more quickly, you can change the duration (in half-seconds) in the lower, right-hand corner of the options dialog, accessible by selecting Input/Options from the main menu.

As a matter of good practice, you should fill out the justification dialog for all entries you make. We won't do so during this tutorial on the software, but any serious use of Constructor will have appropriate and perhaps extensive justification for every entry. You will know which entries have been given justifications and which have not because their input fields will have different colors. By default, the unjustified entries are colored yellow. After you give them justifications, the fields will become white again. Blank entries are professions of ignorance and ignorance doesn't need to be justified, so blanks are always white too. The color coding that Constructor uses works like this:

- Blank white is an invitation to specify,
- Yellow is an admonition to justify,
- White is a justified entry,
- Gray is a consequence of what you know,
- Green is knowledge you didn't know you had, and
- Red is a contradiction between things you thought you knew.

You can turn off this color coding, or change the default colors it uses via the options dialog, accessible by selecting Input/Options from the main menu.

2.1.3 Percentile ranges

Clicking on the Percentiles tab brings up the display in Figure 11. You may also need to click on one of the white input fields and press the Enter key to see the numbers.

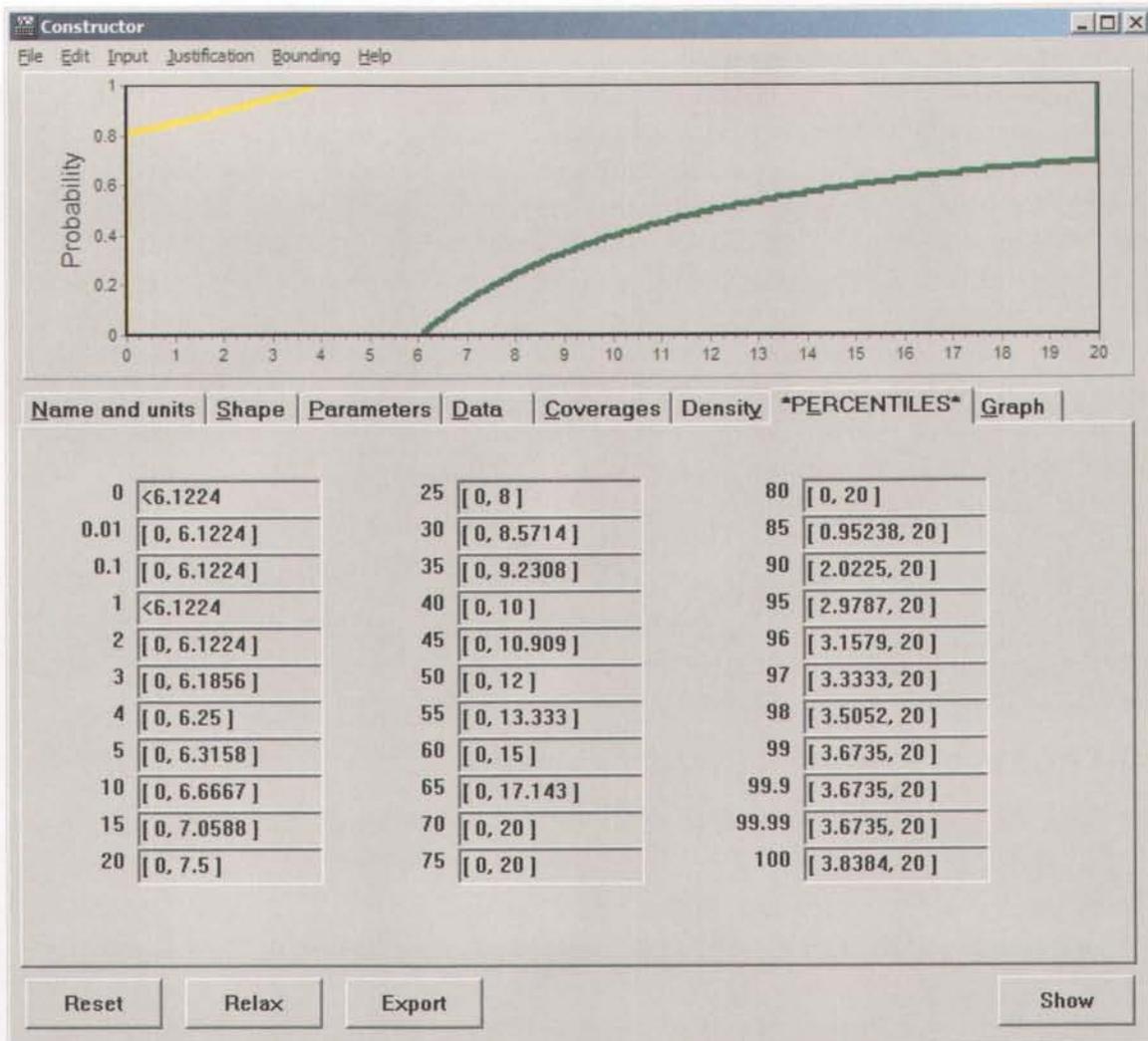


Figure 11.

The percentiles have all been set automatically by the inputs you gave on the Parameters page. That's why they appear in gray. But they are editable too. For instance, entering the better estimate of [4,10] at the 96th percentile cascades both up and down the list of percentiles. Press the Enter key or click out of the input field to make Constructor interpret what you've typed. The result is shown in Figure 12. The p-box also tightens to reflect the additional constraint. The edits that you make to the percentiles will likewise cascade to the other pages.

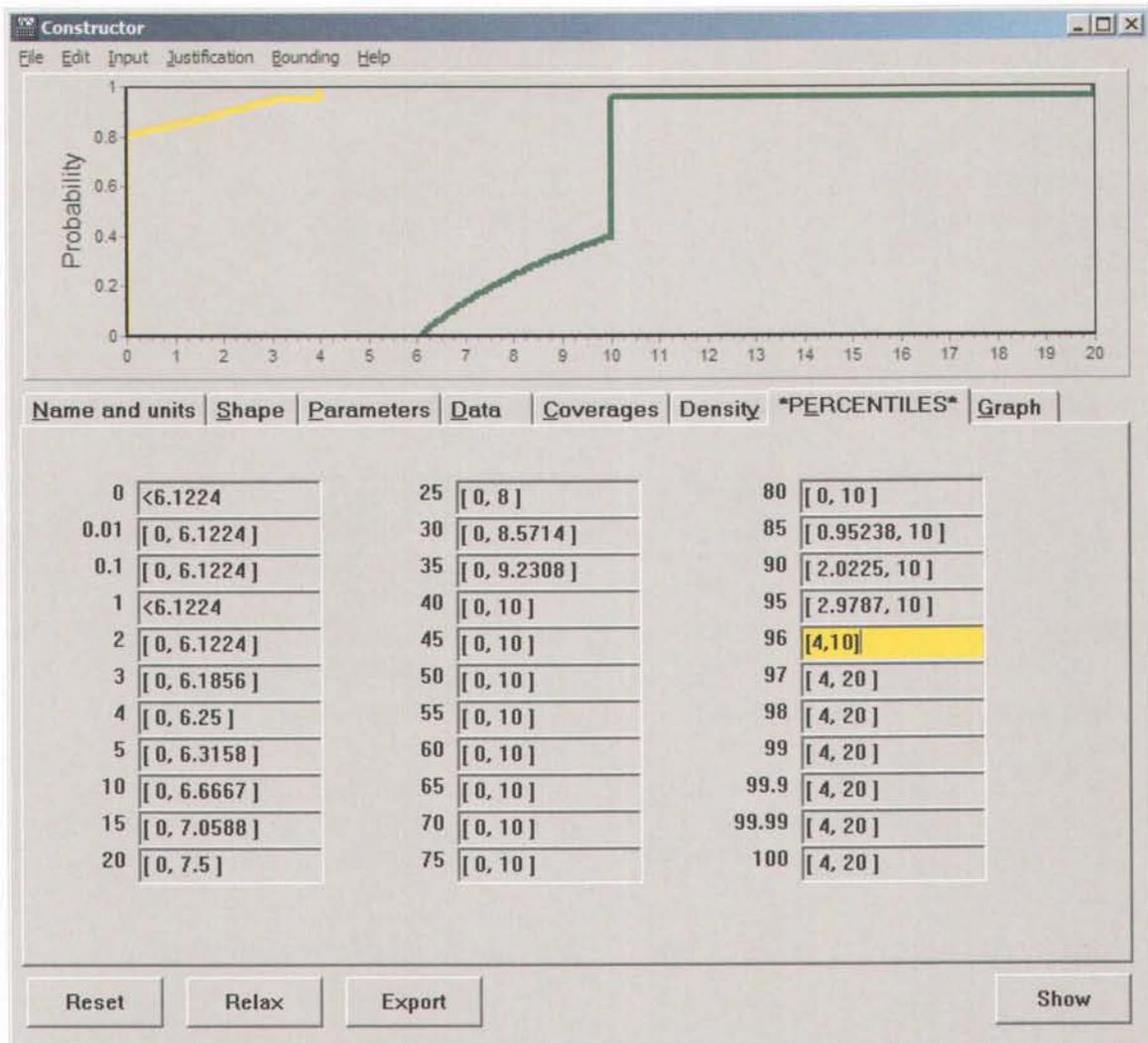


Figure 12.

2.1.4 Graphical limits on the distribution

Clicking on the Graph tab brings up the display in Figure 13, in which you can graphically edit the p-box directly with the mouse.

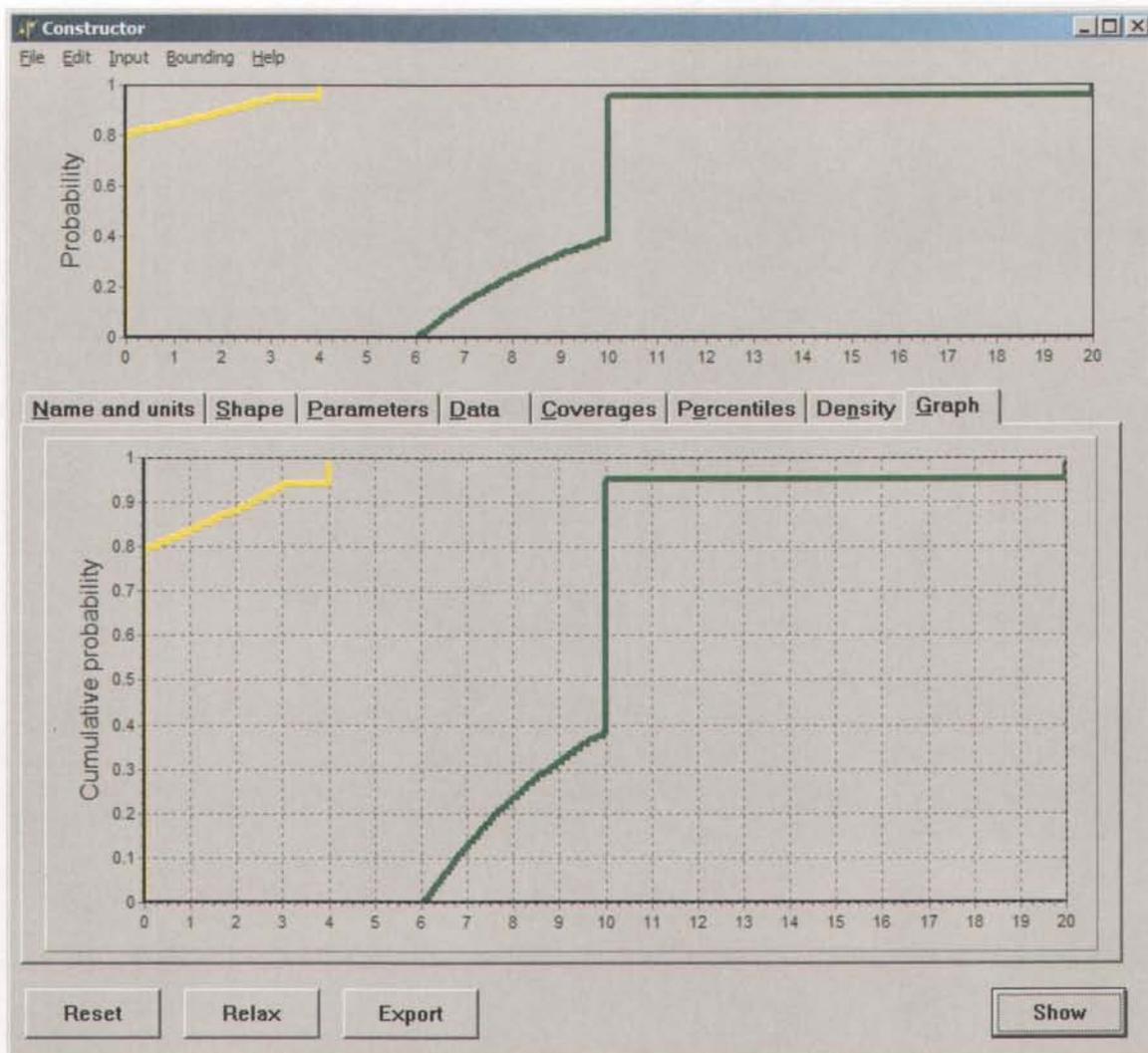


Figure 13.

To graphically edit the p-box, position the mouse over one of the bounds and click to select a point on the bound. You'll know that you have selected a point when the cursor changes from a simple white arrow to a white arrow emerging from a white box (shown at right). While holding the mouse button down, reposition the mouse to where you want the new point to be, and then release the mouse button. The graphs will then be redrawn. It can sometimes be difficult to select a precise point, but, with patience, you can fashion the bounds to arbitrary shapes. You should take care not to allow the bounds to cross each other.



The display in Figure 14 shows the effect of having graphically edited the p-box with the mouse. You could click again on the Percentiles tab to see the numerical effect of this change.

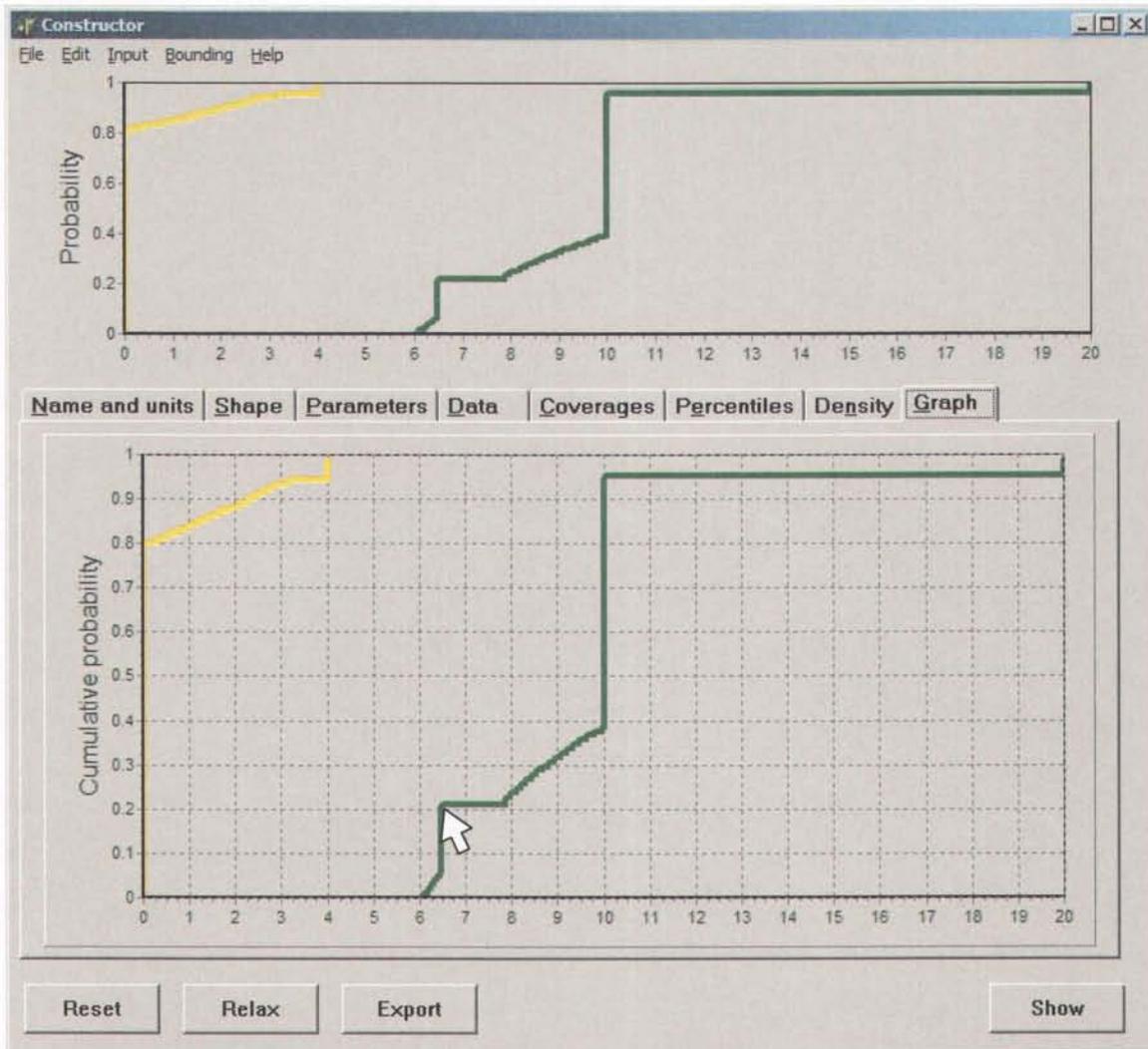


Figure 14.

If you use Constructor a lot, there may come a time when you absentmindedly try to graphically edit the p-box shown in the upper part of the display. The upper graph cannot be edited this way.

2.1.5 Shape constraints on the distribution

Now click on the Shape tab, and you'll see the display shown in Figure 15.

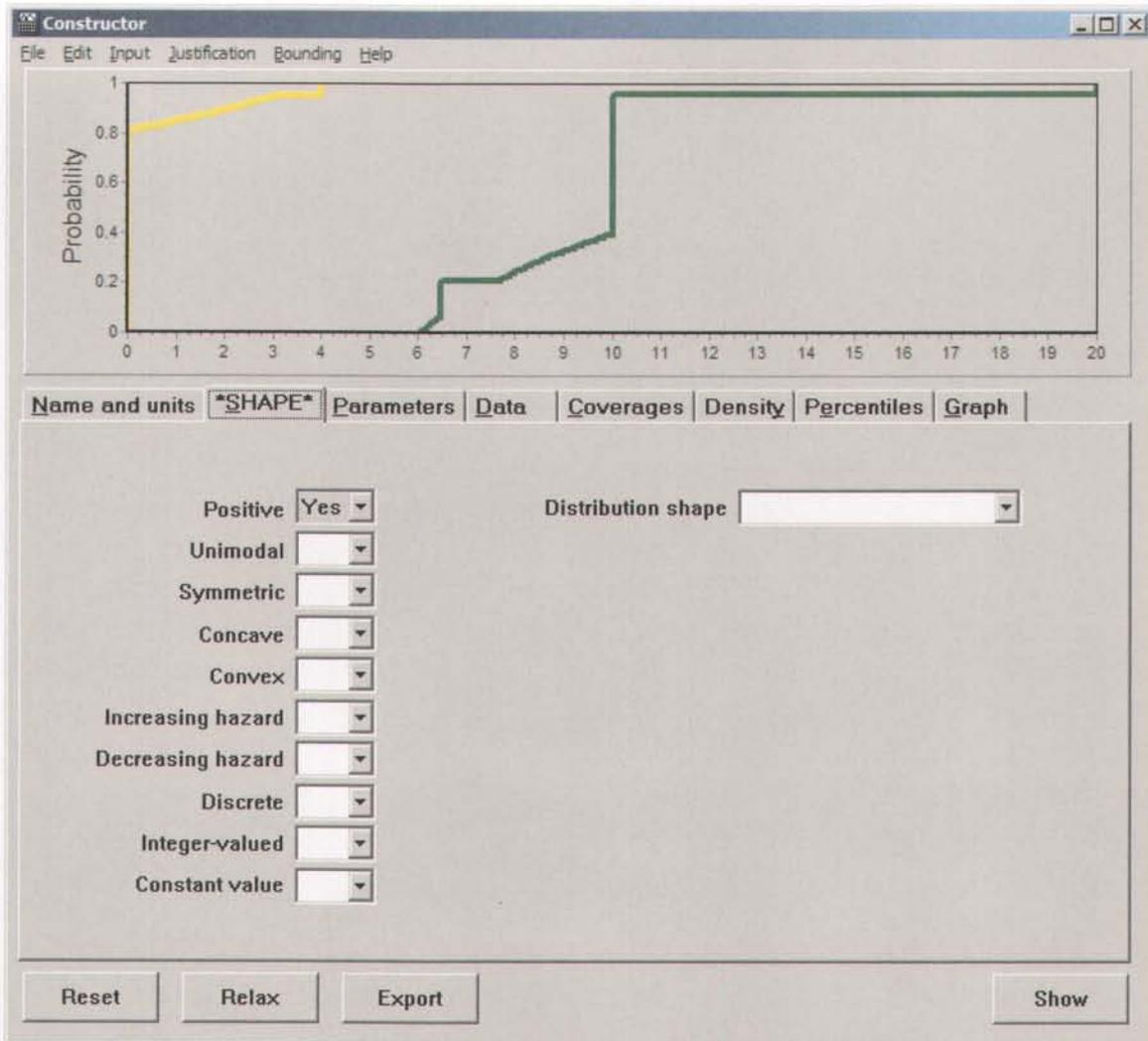


Figure 15.

On this page you can characterize the uncertain number in terms of qualitative features (such as symmetry, convexity, etc.) or by naming a particular family of distribution shapes (such as normal, lognormal, Weibull, etc.).

If you click any of the little white boxes on the left of the page or their adjacent down arrows, you'll see that you can ascribe a Yes, a No or a blank to each of the qualitative features of the uncertain number. The blank indicates that you don't know. If you aren't sure whether or not the uncertain number has a particular qualitative characteristic, you should leave the box blank. Selecting yes or no constitutes a constraint that may narrow the p-box around the uncertain number. We see from the word Yes in the topmost box that the uncertain number has already been inferred to be positive. This was a consequence of our having specified the minimum as zero. Because this was an inference, the box is colored gray.

Suppose that we know that the uncertain number we are characterizing only takes on integer values. Selecting Yes on the second-to-bottom white box makes the display look like Figure 16.

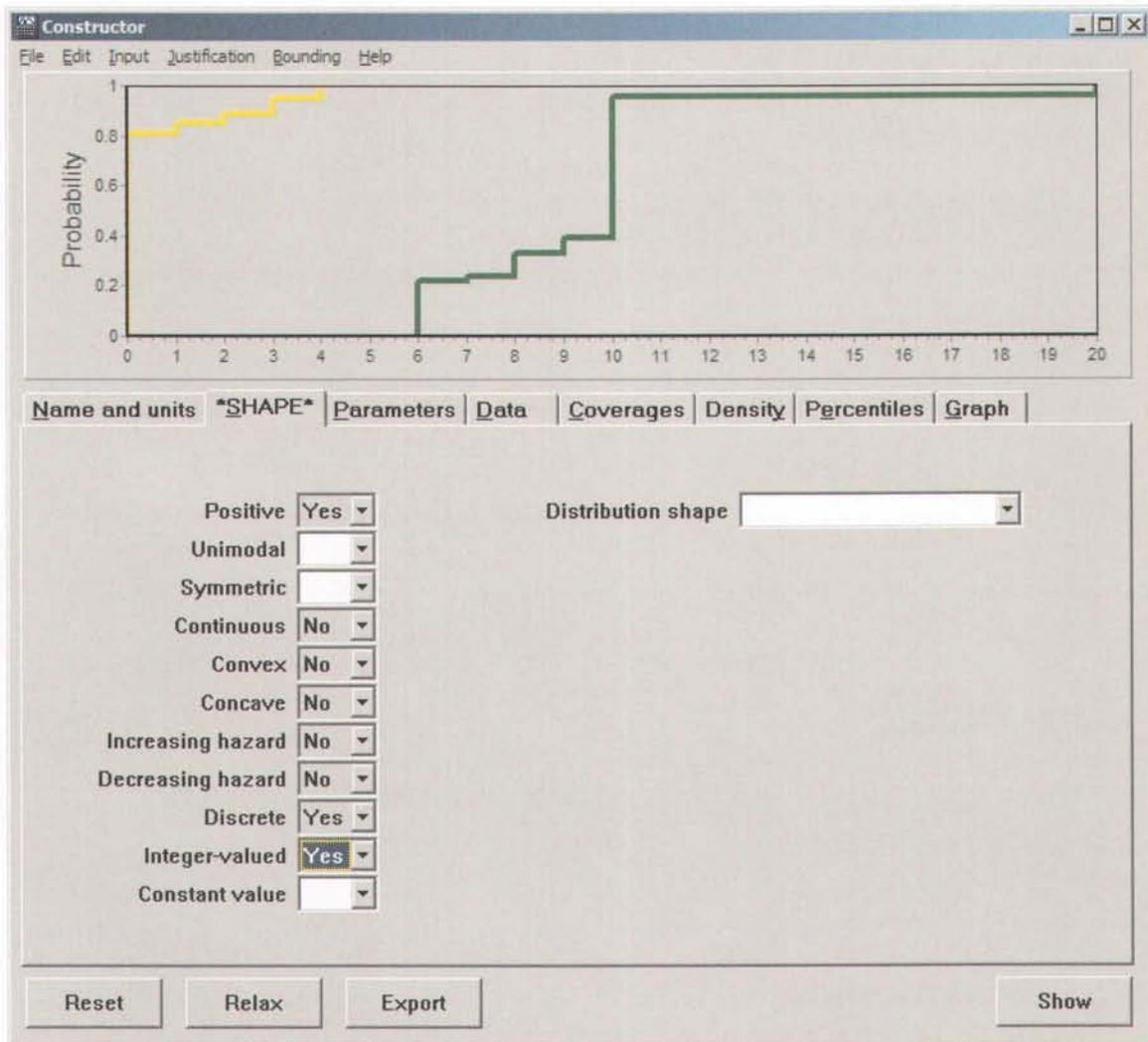


Figure 16.

Notice that the shape of the p-box has been narrowed so that now it only increases at integer values. In addition, settings for several other properties have been inferred.

The box in which you selected Yes has been given a yellow color just like the other direct inputs you have made. (This will become clearer once you click out of the integer-valued box and its contents are no longer highlighted.) Like the other direct inputs, you can make the color of the field white again by giving some justification for the choice you've made that the uncertain number is integer-valued. The justification dialog for doing so is brought up, as it is for other fields, by right-clicking on the yellow Yes. Because this input is not numerical, the dialog is rather different from the other justification dialogs you have seen before. It lacks fields for the magnitude, units and uncertainty, and it doesn't have a little graph.

If you click on the words "Distribution shape" or the white field to their right, you'll bring up the list of possible distribution families. There are about forty choices available on the full list of distribution shapes. In this case, however, most of the distribution families have been disqualified by the assertion that the uncertain number is integer-

valued. The disqualified families are still on the list, but they've been shifted over and prefixed by the word "NOT".

Select "binomial" which is the fourth item on this list by clicking on it. As you do this, the display will look like Figure 17.

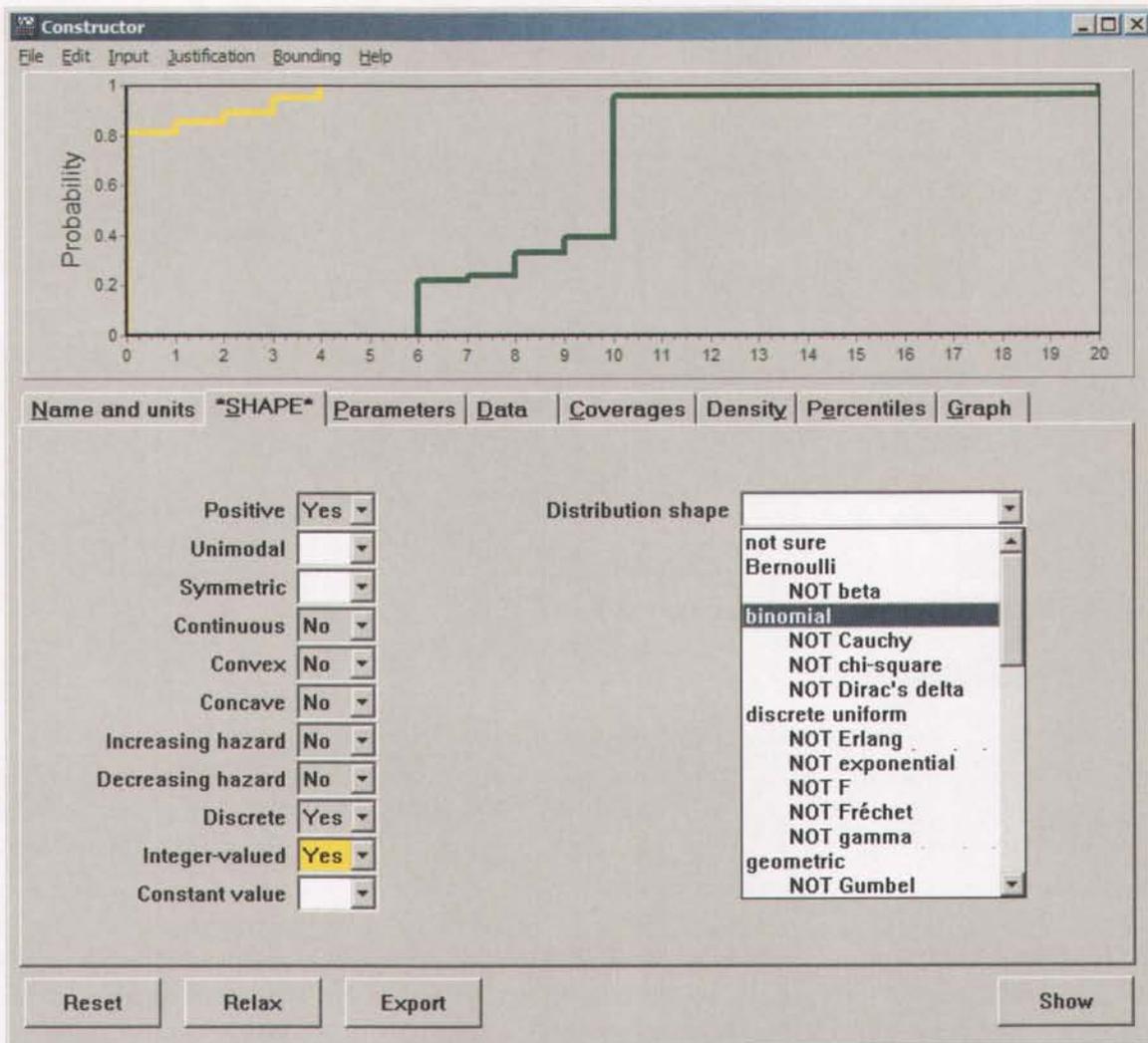


Figure 17.

Picking a distribution family adds input fields onto the page for that family's particular parameters and sets the qualitative shape characteristics on the left to yes or no as appropriate. After you select the binomial family of distribution shapes, the display will look like Figure 18.

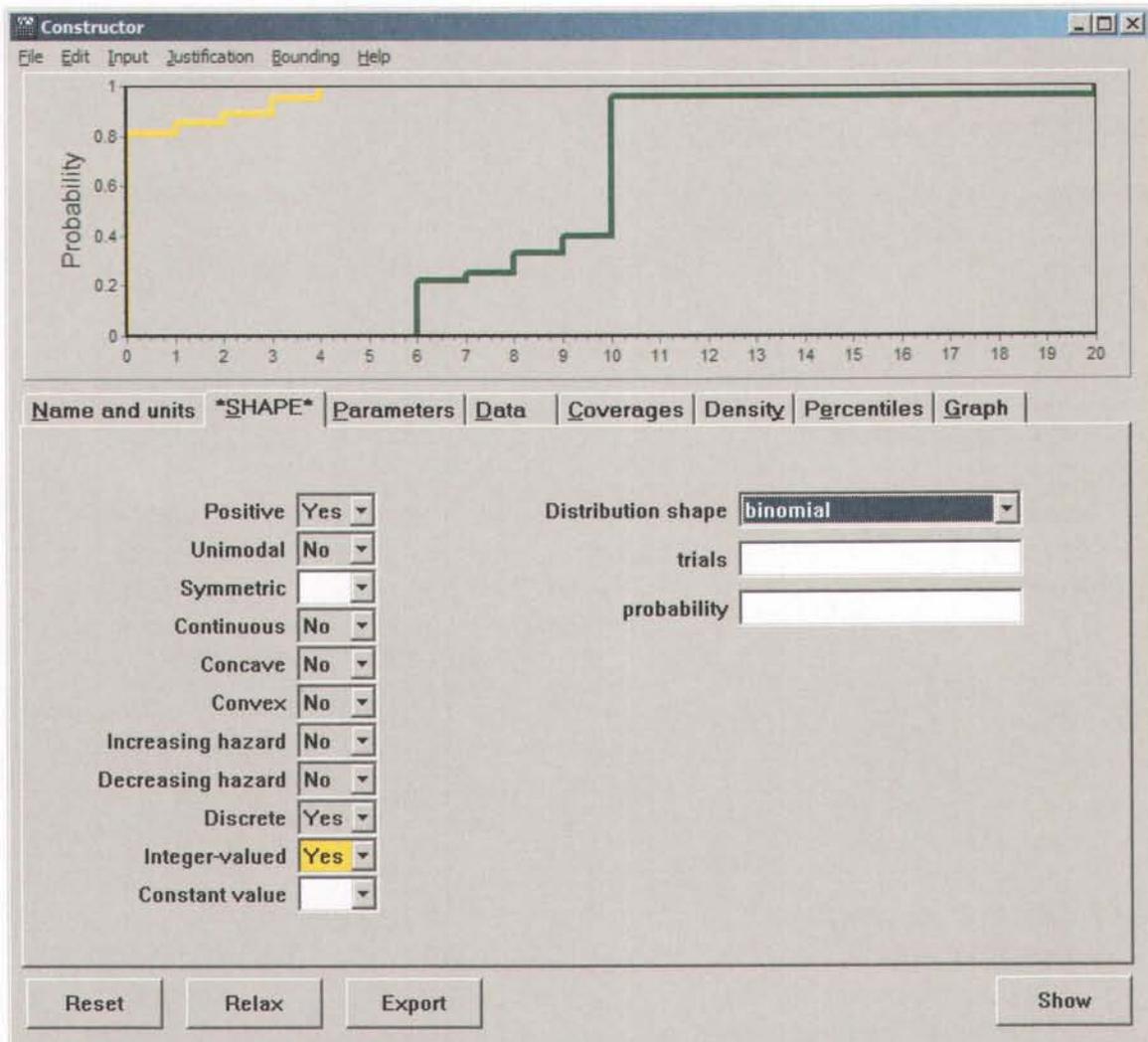


Figure 18.

Selecting binomial as the distribution shape creates two new input fields, one for the number of binomial trials and one for the binomial probability. It also lets Constructor make the further inferences that the distribution is not unimodal, convex, concave, or has a monotone hazard rate. Notice that the choice of the distribution shape is also yellow. As for the other yellow fields, the color reminds you that you should right-click and fill out the justification dialog for this entry.

If we had selected a *normal* distribution, input fields for the mean and standard deviation would have appeared instead. Because Constructor already knows intervals bounds on those parameters, they would have been filled in automatically. However, because normal distributions cannot be integer-valued (except in the degenerate case where its standard deviation is zero), there would have been a conflict with the choice we made about that qualitative shape feature. Normal distributions can't be entirely positive either (because of their infinite tails), so both the positive and integer-valued fields would appear in red to indicate the contradictions between normality and constraints we have already described.

Enter the interval $[3, 20]$ for the number of trials and the interval $[0.2, 0.4]$ for the probability and press the Enter key. The display will then look like Figure 19.

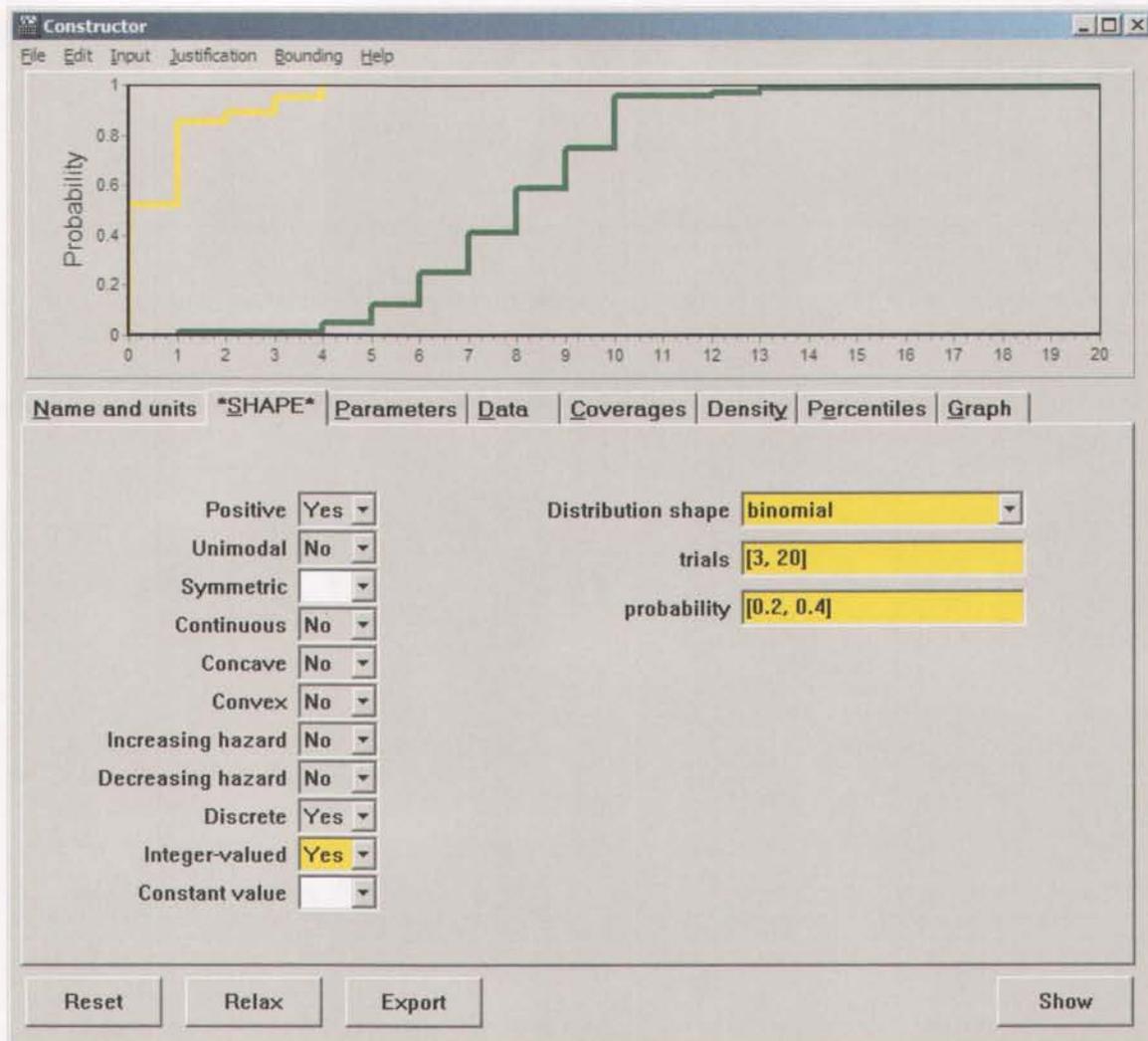


Figure 19.

The p-box at the top of the display got somewhat tighter. If the parameters we gave had been precise scalars rather than intervals, the result would have been a precise distribution.

2.1.6 Density bounds

You can also specify upper and lower bounds on the probability *density* of the uncertain number. You can access the density bounds editor by clicking on the Density tab. The display in Figure 20 has already had some inputs. If you want to create a similar display on your computer, select File/Open from the main menu, enter the file name DENSITY.CON and click the Open button.

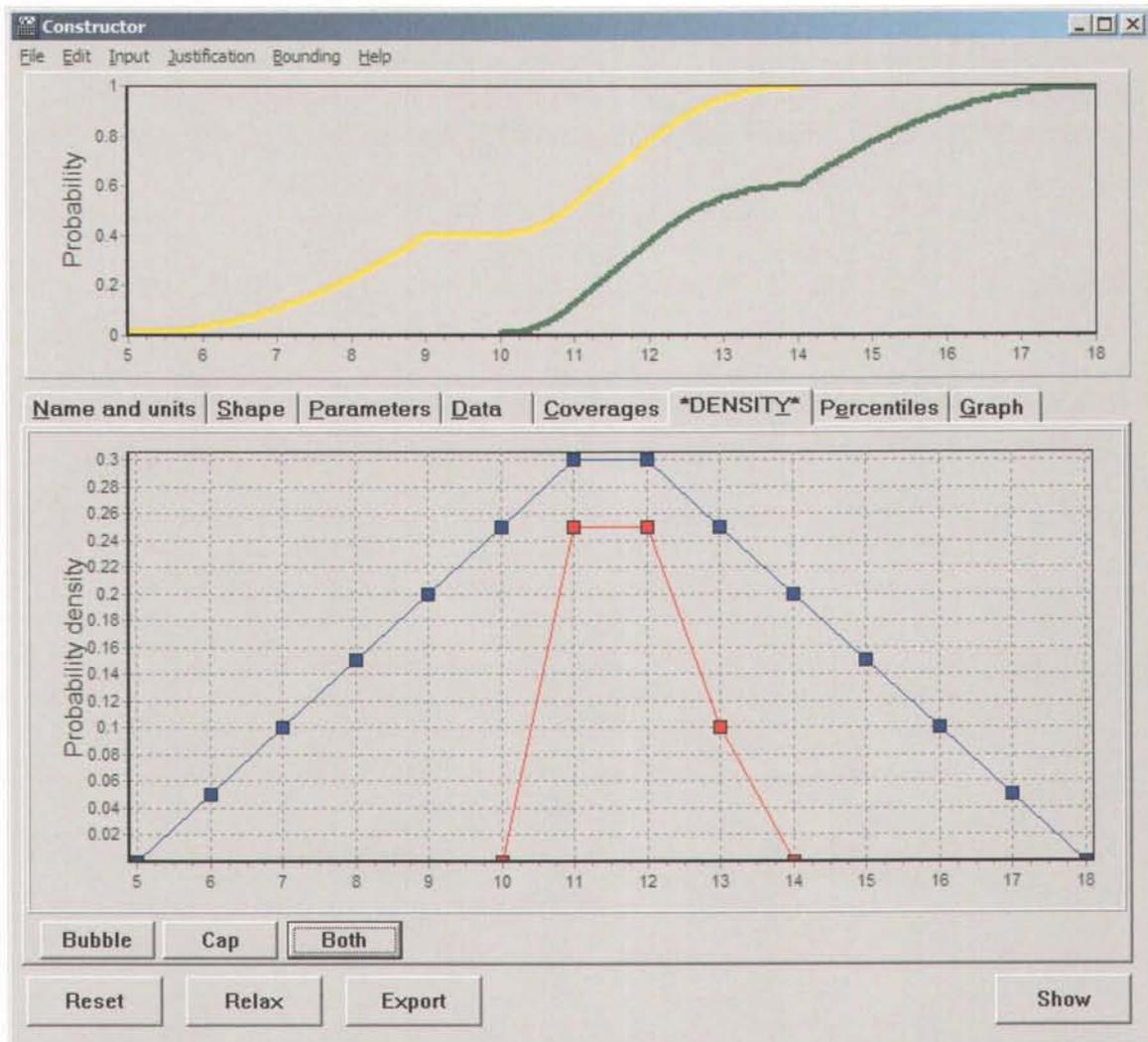


Figure 20.

The blue points and the lines connecting them describe the upper bound on the probability density function of the uncertain number. This is called the “cap”. The red points and their interconnecting lines represent the lower bound on the probability density function. We call this the “bubble”. The cap constrains the density from above and bubble constrains it from below.

To be interpretable as an upper bound on density, the area beneath the cap must be larger than one. (It could equal one, but if it did, it would perfectly prescribe the distribution function irrespective of that the bubble was.) Likewise, the bubble would have area less than one (equal to one would specify the distribution precisely). The bounds on the cumulative distribution function that follow from the specified bubble and cap are displayed in the upper graph of the display. The algorithm to compute these bounds is described by Ferson et al. (2003). As the bubble and cap converge, they more narrowly prescribe the distribution function. Obviously, the bubble and cap should never cross each other, because this would be nonsense.

Constructor allows you to use the mouse to graphically edit the bubble and cap bounding the density function. To move a point, click on it and, while holding the mouse

button down, reposition the mouse to the spot where you want the point to reside. Put it there by releasing the mouse button. With each edit, the p-box in the upper graph will be recomputed. If your edits cause the cap to have area less than one, or the bubble to have area greater than one, or, horrors, the bubble and cap to cross, you'll get an error message.

Of course, you can edit the density constraints numerically too. Just double click on the (lower) graph to invoke little spreadsheets in which you can specify the points of the bubble and cap by entering numbers. The display in Figure 21 below illustrates this spreadsheet interface.

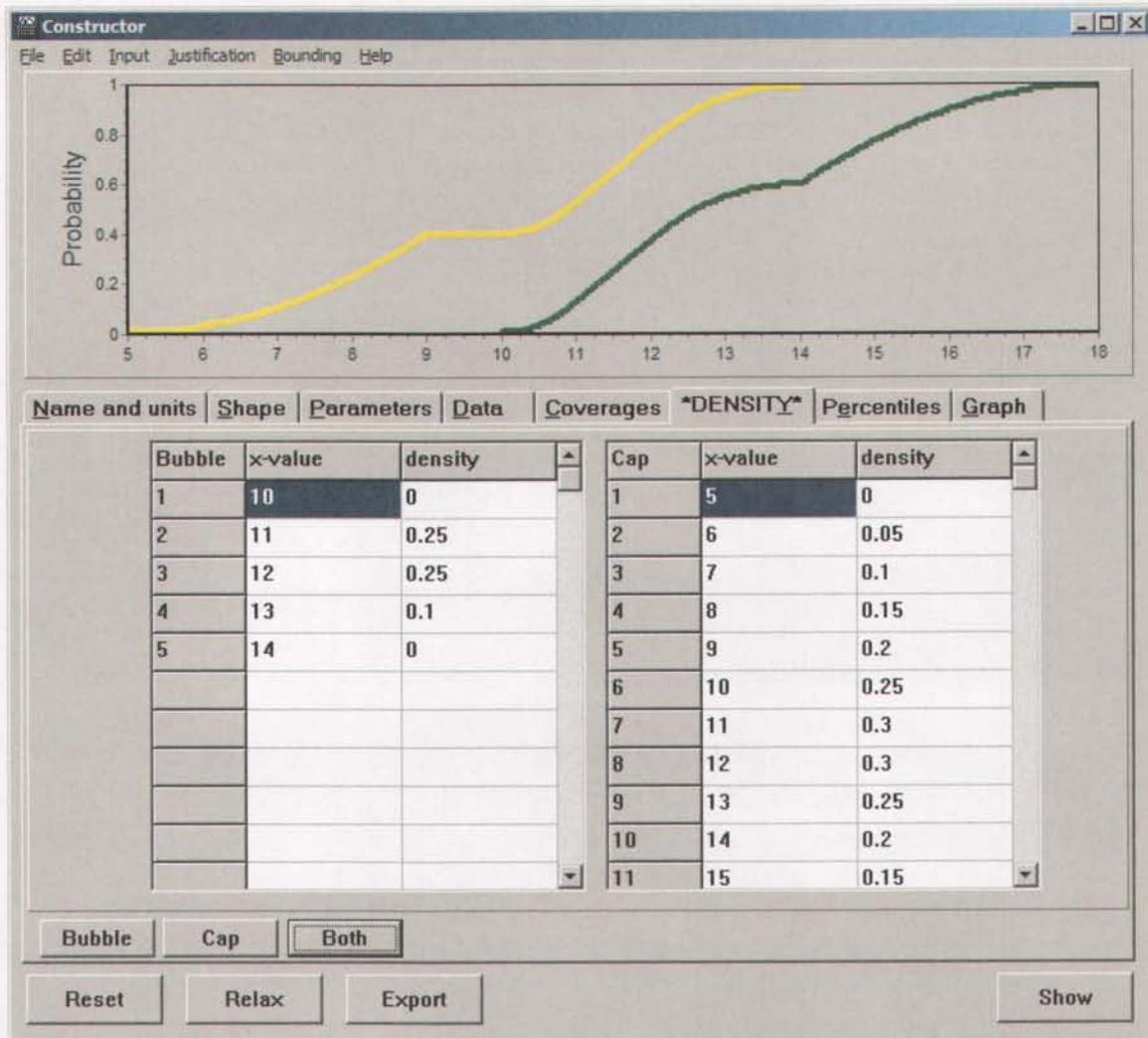


Figure 21.

As you change any of the numbers in the spreadsheets, the p-box in the upper graph is recomputed. The x-values should be given in ascending order. If they're not, a warning message will appear at the bottom of the page. A linear function will be interpolated between the points you specify.

You will probably find it easiest to *start* entering points on the Density page with the spreadsheet interface. Double click on the lower graph (which will initially be blank) to access the spreadsheets.

If there is no function displayed over a value of the abscissa of the lower graph, the function is assumed to be zero there. Thus, the bubble is zero below 10 and above 14. This means that the probability density of the uncertain number could be as low as zero below values of 10 and above values of 14. (For intermediate values, the density could not be so low.) Likewise, the cap is zero outside of the interval [5, 18], but the meaning of these zeros is very different for the cap. It means that the density certainly must be zero beyond this interval. Because Constructor draws straight lines between adjacent points, if you want the bubble or cap to be zero someplace *within* the range of numbers, you would have to explicitly specify the values where the bound is zero.

The information embodied in the bubble and cap can be used to tighten most of the parameter estimates. The median, minimum, maximum, interquartile range, and range can be improved, although the constraints are only those evident from the constraints on the p-box from the bubble and cap. In addition to constraining the p-box, the bounds on the density can also directly inform the mode, mean, variance, and standard deviation. For instance, clicking back to the Parameters page reveals that the mode must now be between 10 and 13. Can you see from the graph at the bottom of the Density page why these are limits on the mode? These bounds on the mode could not be derived if we had only the p-box depicted on the upper graph of the display.

It is interesting that, although bounds on the density (i.e., the bubble and cap) can be used to derive bounds on the cumulative probability (the p-box), the reverse is not true. All that knowing a p-box tells us about the bubble and cap is their possible ranges. Unless the sides of a p-box happen to coincide at a point, the cap at that x -value can be any value between zero and infinity, and, likewise, the bubble can be any value (smaller than the cap) ranging between zero and infinity.

2.1.7 Coverage probabilities

You can also specify some interval ranges within which the uncertain number is known to lie with a certain probability. Load the file CLEAR.CON to erase the previous inputs. When you click on the Coverages tab, the display should look like Figure 22.

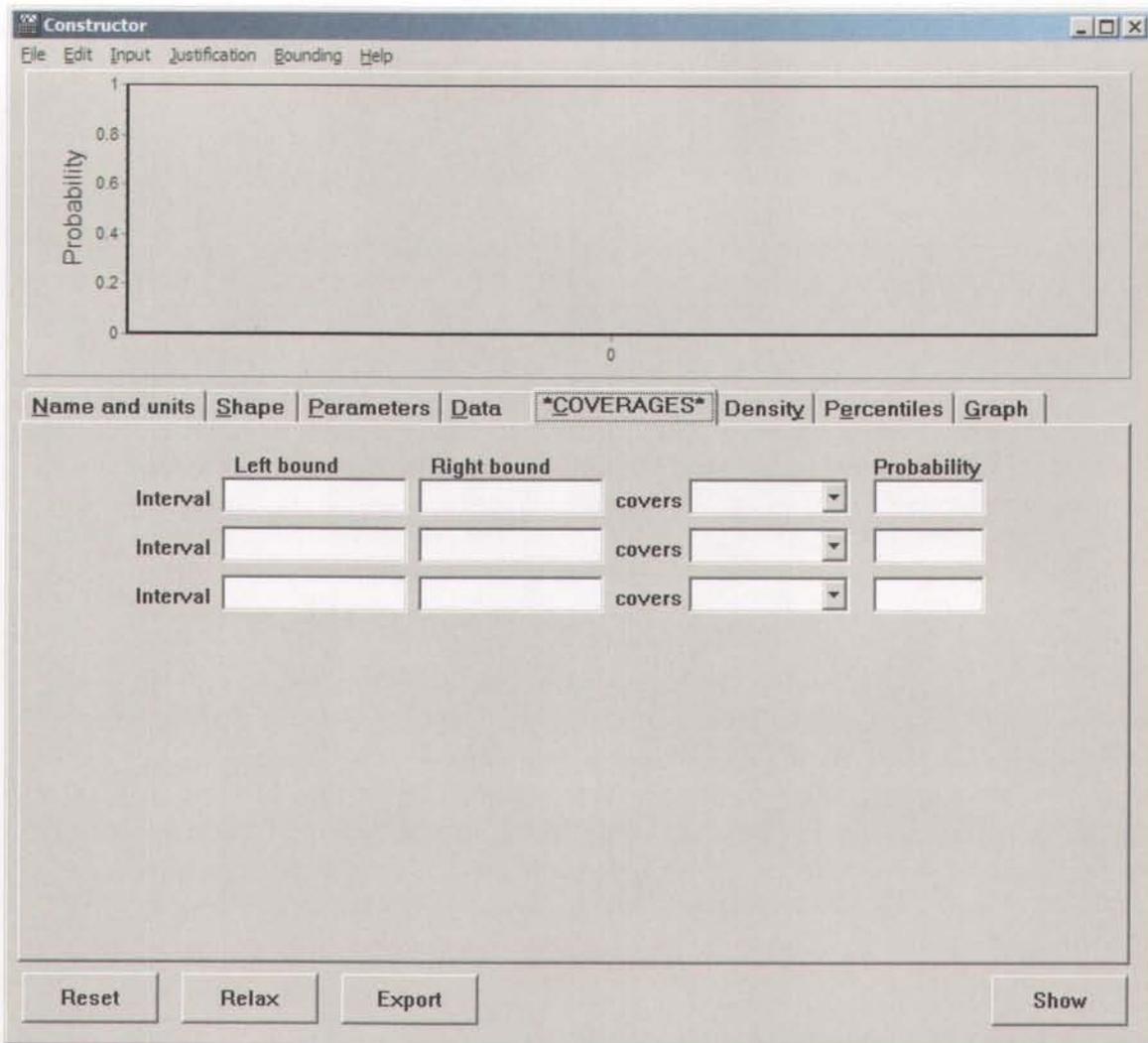


Figure 22.

The page initially shows three rows for inputs. In each row you can enter the left and right bounds of some interval where the uncertain value is known to be sometimes. For each such interval you enter, you should also indicate in the fourth field of the row the probability that the uncertain number will have a value inside that interval. By clicking on the third entry field of each row, you can select “exactly”, “no more than” or “no less than” to describe how the probability relates to the interval. Failing to select one of the three choices is equivalent to having selected “exactly”. As you fill in all the rows with probabilistic statements, a new row for inputs becomes visible. Up to ten rows can be used to characterize coverages for the uncertain number.

The probabilities need not add up to one and, typically, they won’t because they represent coverages that constrain the same probability mass in different ways. However, because they are probabilities, no value can be smaller than zero or larger than one. You can specify the probabilities as precise scalar values or as intervals. Use an interval when you are unsure about the probability to use.

You can specify the potential range of the uncertain number by giving an interval and indicating that it covers with probability no less than or exactly equal to 1. If you

have such a statement, it will be convenient to enter it first in the top row on the Coverage pages. There should be at most only one such statement.

Each right bound must be no smaller than the corresponding left bound. If they are the same value, you are saying that some portion of the probability mass is known to lie at that value. It is possible to enter logically inconsistent statements. For instance, if you say that the interval [1,2] has 60% of the mass and that [3,4] also has 60%, you have created a logical impossibility. (Note, on the other hand, that such probabilities would be entirely possible if the intervals were [1,3] and [2,4].) Because it is easy to inadvertently make mutually inconsistent specifications, you should be careful about the inputs you make. Such inconsistencies may result in the upper and lower bounds crossing each other. If this happens, the software will give you a warning. However, it is also possible that an inconsistency does not result in the bounds crossing.

There are three kinds of coverage statements that can be made. They are constraints on a random variable X of the following forms.

$$\begin{aligned} P(X \in [x_1, x_2]) &\geq [p_1, p_2] \text{ (“no less than”)}, \\ P(X \in [x_1, x_2]) &\leq [p_1, p_2] \text{ (“no greater than”)}, \text{ or} \\ P(X \in [x_1, x_2]) &= [p_1, p_2] \text{ (“exactly”)}. \end{aligned}$$

where P denotes the probability of an event and the subscripted symbols denote the numerical values entered on the Coverages page. Note that, for a “no less than” statement, the lower bound on the probability is not controlling, so only p_2 is used. Likewise, for a “no more than”, only the lower bound probability p_1 is important.

Given a suite of such statements, the calculation must find upper and lower bounds on the cumulative probability distribution function for X . This is a linear programming problem. For each bound on probability at every possible value of X , there is a set of equations and inequalities that must be satisfied. The objective function is to maximize or minimize the probability at that x -value. Following Kozine and Utkin (2004), the set of all x -values is the range from the smallest to the largest x -values mentioned in any of the intervals of the constraints. The procedure used here extends their work by allowing for all three types of (in)equalities, provides error checking for impossible sets of constraints, and is also valid for cases where the coverage intervals are not nested in both x and p .

Figure 23 depicts an example of the output produced by three coverage statements.

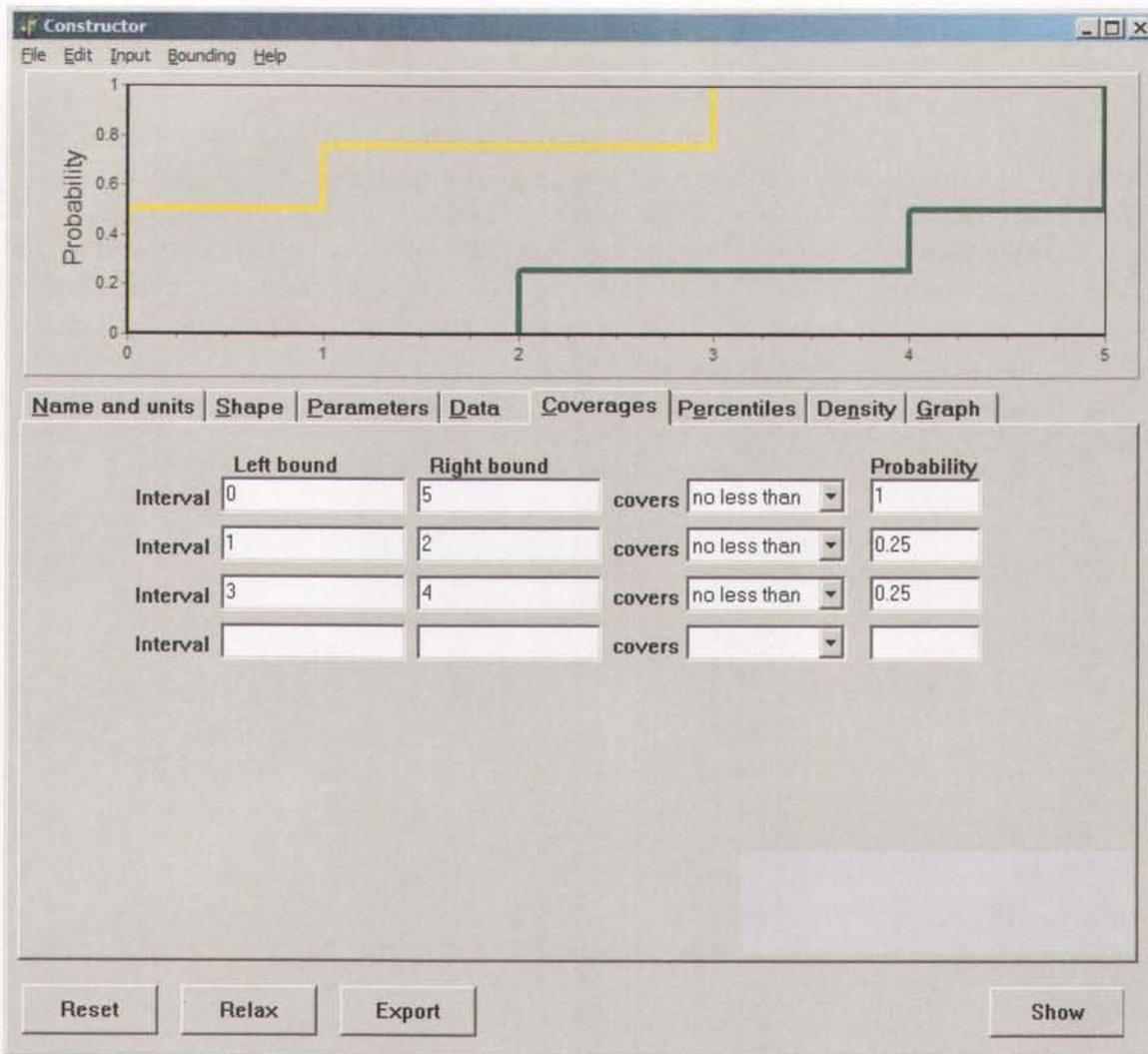


Figure 23.

As you replicate the entries on this input page on your computer, the fields will turn yellow to remind you that inputs need to be justified. You may notice that the p-box at the top of the display changes with individual keystrokes, rather than only when you press the Enter key or tab out of an input field. When you fill up the third row of inputs, a fourth row will appear. You can have up to ten rows.

2.1.8 Data

The last tabbed page of inputs to review in this tutorial is the Data page. We consider it last because, in many cases, actual sample data is the rarest kind of information. Moreover, the treatment of sample data and the inferences to be drawn from them are in many ways the most controversial issues among the methods implemented in Constructor. The section “Handling imprecise data” on page 66 reviews some of these issues and describes several possible strategies. For now, we’ll just illustrate a few of the ways you can wrangle data with Constructor.

Click on the Data tab to bring up the Data page. Then select Bounding/Exploratory from the main menu. This puts Constructor into exploratory mode, which changes its

behavior in several ways (see page 41). It turns off the automatic cascading of inputs to other input fields, which lets you make entries and change settings without having to worry that they are consistent with the previous inputs. Exploratory mode also makes some sample data sets available for your review. A new button labeled “Test” has now appeared in the lower, right-hand corner of the page, above the Show button. Clicking this button creates a small sample data set and produces the p-box that characterizes the data set. After you click the Test button once, a small number control appears next to the button. This is the test data set number. You can edit the number by typing in the field or by clicking the small up and down arrows between the number and the Test button. If the number is zero, the display will look like Figure 24.

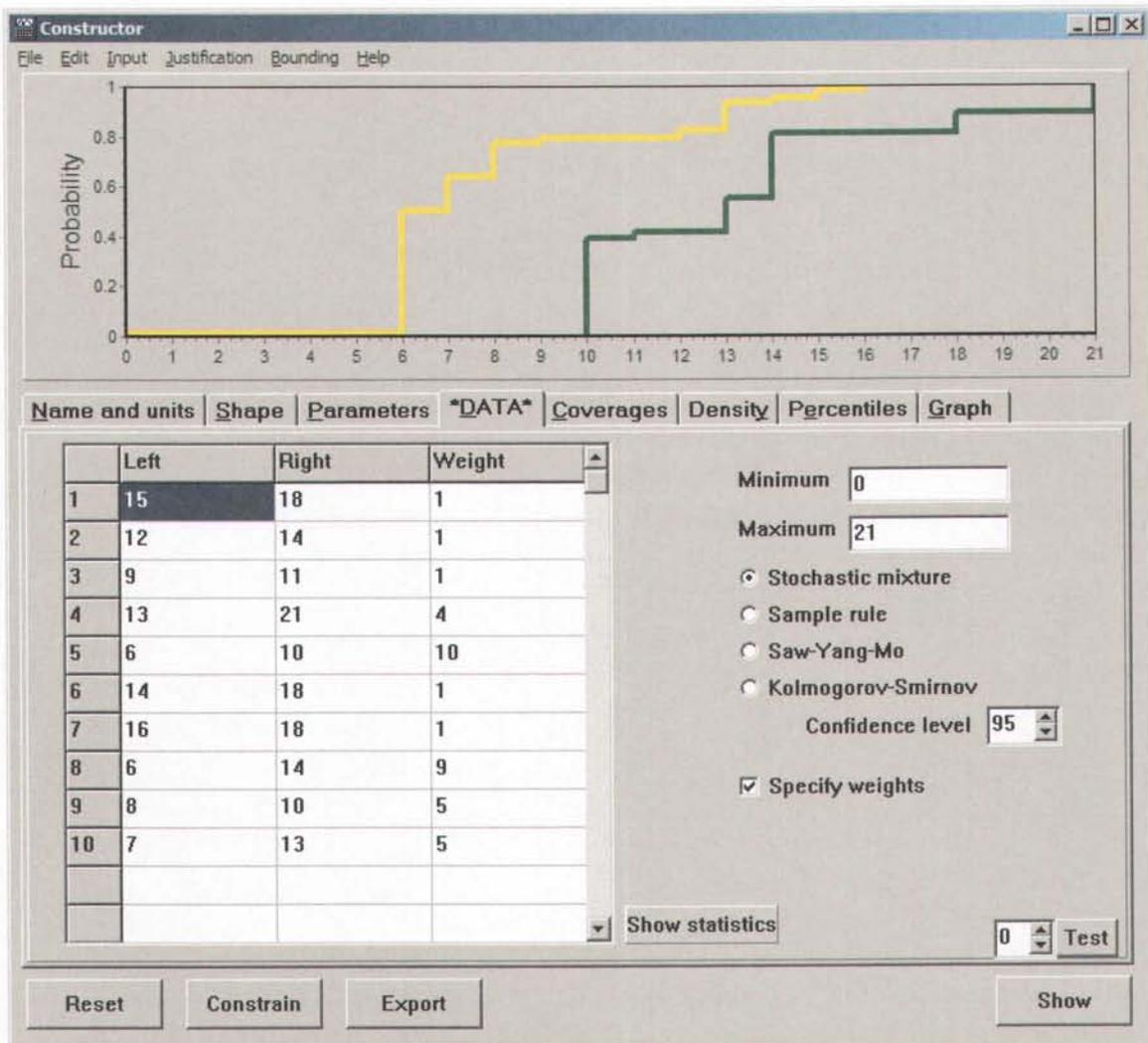


Figure 24.

This display shows a data set characterized as a stochastic mixture (see page 66). This is one of four methods for characterizing the uncertain number from sample data that are available on this page. You can choose between these methods by clicking on one of the four round radio button on the right side of the page.

In this case, the sample data set consists of ten weighted intervals as shown in the spreadsheet on the left of the page. The left bound of each interval is in the first column and its right bound is in the second column. The weight for the interval is in the third column. The left bounds must be less than or equal to the corresponding right bounds

Each interval represents a sample value taken from the population specified in the ensemble field on the Name and units page. The width of each interval represents the measurement uncertainty about that sample value. You can enter scalar data, but you would have to enter them as degenerate intervals by letting the right bound and the left bound be the same value. Only in rare cases, however, would empirical data actually have no measurement uncertainty, which is what scalar data would represent.

Uncheck the box labeled “Specify weights” (by clicking on it). When you do the p-box at the top of the display will change rather dramatically. The display will then look like Figure 25.

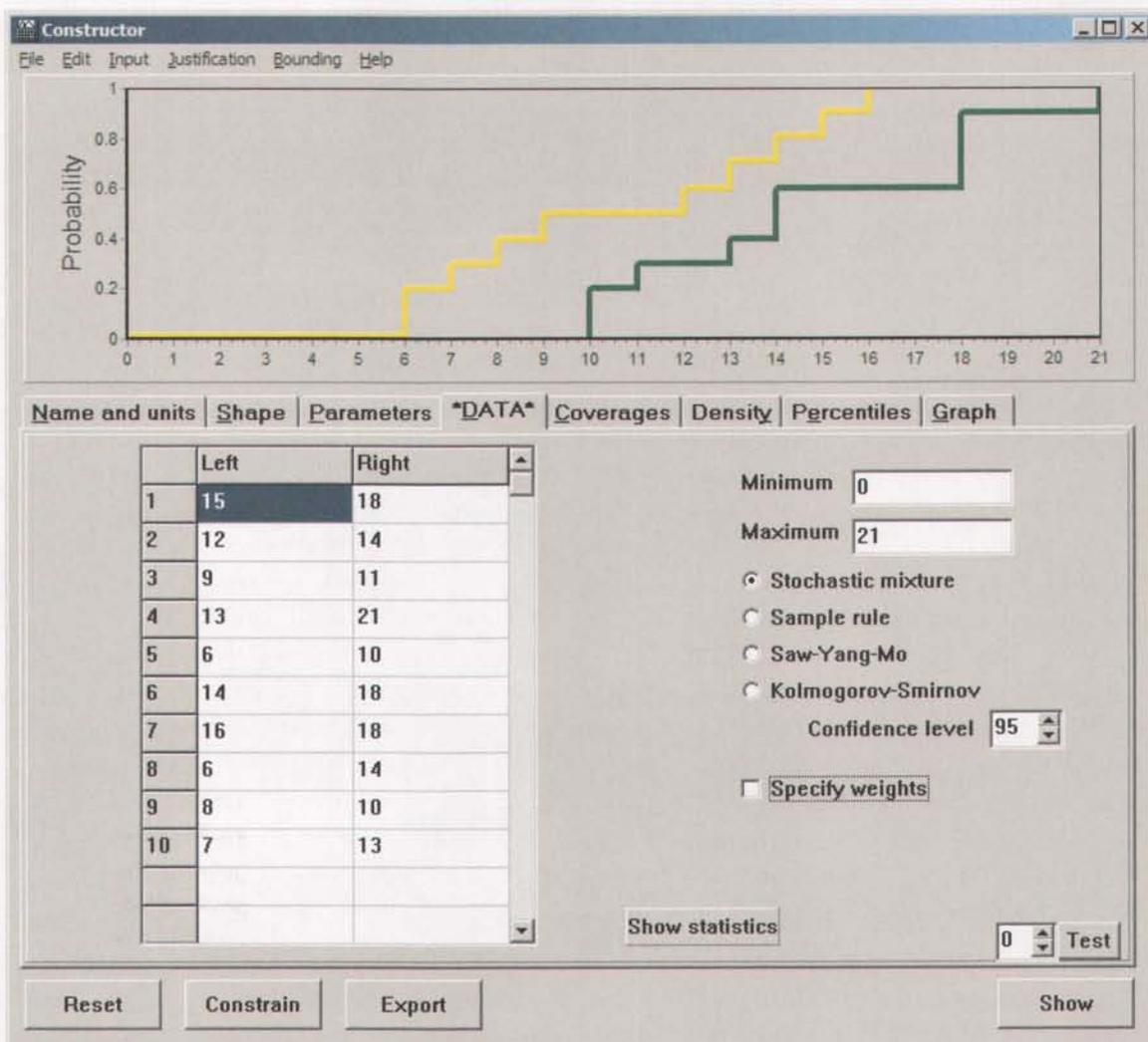


Figure 25.

The difference arises because the previously shown p-box used the weights in the third column of the data spreadsheet and this p-box presumes that the ten input intervals are all

equally weighted. You would see the same p-box if you changed all the weights in the third column to be ones (or any other constant positive value) and double-clicked on the stochastic mixture radio button to recompute the result.

The weights should all be positive values. Constructor permits two interpretations for these weights. They could represent simple poolings of like-valued samples, or they could represent transformations intended to scale the impact of some samples relative to the others. The former interpretation would be useful in processing very large data sets that would otherwise have too many points for Constructor to handle. The latter would be useful in accounting for non-randomness in the sampling that created the data. For instance, suppose each sample represents a spatial region but the samples were not randomly located in space but rather concentrated in hot spots where the values are largest. A spatial weighting that partially corrects for the non-randomness of such samples could be fashioned by using the area of the Thiessen polygon associated with each sample point. Specifying the areas as the weights would give less the impact of clustered samples and increase the impact of geographically isolated samples. In the first interpretation of weights, the number of degrees of freedom associated with the data set is a function of the sum of all the sample weights. In the second interpretation, the number of degrees of freedom is a function of the number of sample intervals, irrespective of their weights. You can select which interpretation should be used for the data set by checking the appropriate radio button in the “Weights” portion of the options dialog accessible by selecting Input/Options from the main menu.

Now click the small up arrow immediately to the left of the Test button. This will make the number 0 change to a 1 and another sample data set to be displayed. This data set has also ten different intervals, but there are no weights displayed given for these intervals. The p-box constructed from the data set is based on the relaxed sample rule described on page 69. If you click the “Show statistics” button at the lower, right-hand corner of the data spreadsheet, a window with several summary statistics for the data set will appear on the right side of the page. The display will then look like Figure 26.

The summary of the statistics can be a useful tool to check whether all the data was copied into Constructor and are being interpreted correctly. Is the sample count what you expected? Is the mean correct?

There are a dozen more input data sets you can review by changing the number next to the Test button. The first four data sets are fixed, but the later data sets are generated partially with random numbers, so their details can change each time they are displayed. You can also enter your own data on the spreadsheet. See page 52 for advice on doing this. After entering your data, you can get Constructor to compute the p-box for the data you've entered by pressing the Enter key while editing the spreadsheet, by changing which of the method radio buttons is checked, or by double-clicking on the radio button of the method you want to use to construct the p-box.

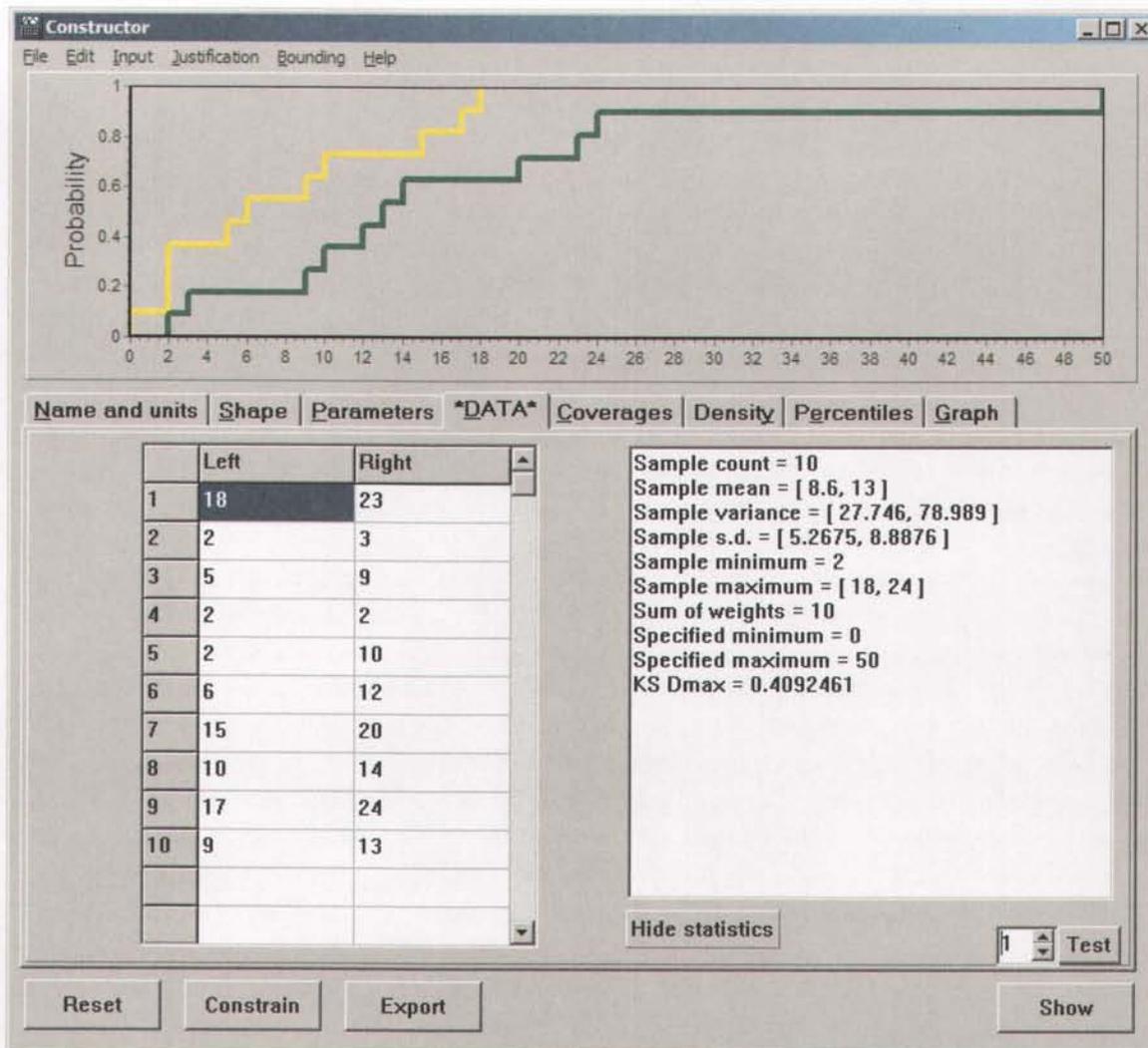


Figure 26.

Because Constructor is in exploratory mode (see page 41), a Test button now appears in the lower, right-hand corner of most of the tabbed input pages. You can click these buttons to see sample input sets designed to illustrate Constructor's functionality for each input page individually. Repeatedly clicking Test selects at random from several pre-selected input sets. You can also select the sets numerically by changing the number that appears to the immediately left of the Test button after you click it the first time.

Exploratory mode alters Constructor's behavior in several ways. You may want to leave Constructor and re-enter the software to restore its functionality before attempting to use it to work on a real problem. If you want to continue with the software tutorial in the next section on justifications, you may want to restore automatic color-coding of input fields which exploratory mode disabled. To do this, select Input/Options and check the box labeled "Color code input fields" on the dialog that appears. Click the Done button to close the dialog.

2.2 Exploratory mode

By default, Constructor assumes you actually want to construct a distribution or p-box from available data. Sometimes, however, especially when you are first becoming acquainted with the program, you just want to see what options are available and browse through its features. Constructor offers a special exploratory mode to make this easy.

You can invoke exploratory mode by selecting Bounding/Exploratory from the main menu. When you do this, automatic cascading of the inputs to other fields is turned off. This will suppress most of the error and warning messages that would otherwise be generated when you make inputs that don't agree with each other. In exploratory mode, you can change your mind, back up, and arbitrarily alter inputs as you like (rather than only narrowing them).

Entering exploratory mode also changes the bounding strategy so that the p-box depicted in the upper graph of the display is based only on the information in the current tabbed page of inputs. This allows you to switch among the tabbed pages and explore the features of each separately.

Exploratory mode turns off the color coding of the input fields, so you won't be distracted by color changes. The sequence of how the input fields have been altered is still recorded in the history fields on the justification dialogs for each input, but there would be little need to provide justifications for any inputs when you are exploring.

While exploring in Constructor, it may be useful to turn on echo plotting, a feature of Constructor in which the previous graph on the upper part of the display appears along with the current graph. The previous graph is depicted in thin lines. Echo plotting allows you to visualize the effect of any change you make on the input or program settings. You can turn on echo plotting by selecting Input/Options from the main menu and checking the box at the far right side of the options dialog labeled "Show echo of previous plot". Echo plotting works when you change input on a page and even when you switch pages. Clicking on the Show button at the bottom, right corner of the display redraws the current graph and thus works to erase the echo plot. Turn off echo plotting by unchecking the box on the options dialog.

When you first enter exploratory mode, new buttons labeled "Test" will appear in the lower, right-hand corners of most of the input pages. Clicking on the Test button will generate and depict some example inputs for the current input page. These examples show some of the various kinds of statements you might make and the p-boxes that would result from such statements. After you click the Test button on a page, a little counter appears next to the button that shows the number of the example data set. Clicking on the Test button again picks at random from the list of examples. You can also select a particular data set by clicking the up and down arrows on the counter or otherwise entering the number of the data set you want to see. Cycling through all the available example data sets on each of the input pages constitutes a fairly comprehensive introduction to the basic features of Constructor.

In exploratory mode, and whenever automatic input constraining is turned off, the Percentiles page behaves quite differently than it normally does. To see how, first click the Percentiles tab to go to that page and press the Test button on the lower, right corner of the page. If it's not already there, put a "0" in the counter next to the Test button to select the zeroth test data set. Most of the white fields for percentiles are empty except

for four entries given by the example data set. The graph at the top of the display looks like Figure 27.

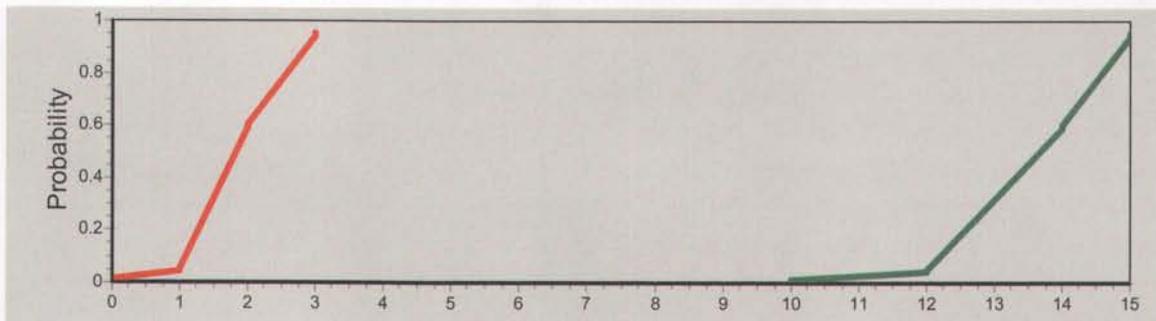


Figure 27.

If you then select Bounding/Constrained from the main menu and click on the zeroth percentile field and then on the Show button, the graph will look like Figure 28.

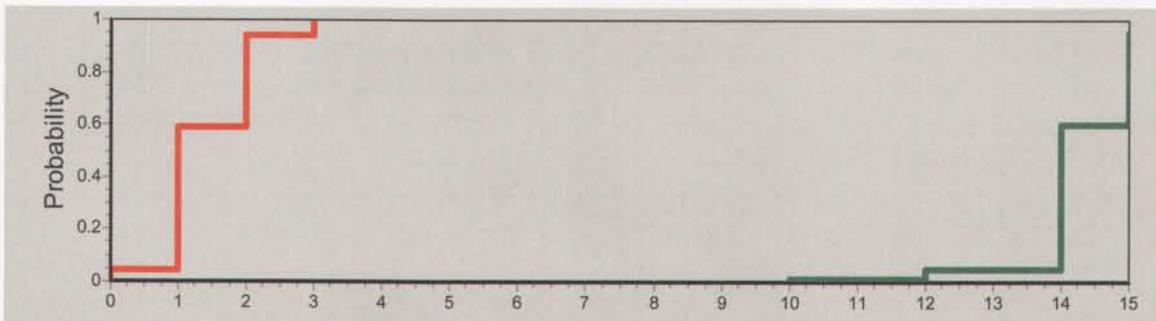


Figure 28.

The relationship between these two plots is easiest to understand if echo plotting is still turned on. If it is, the graph will look like Figure 29.

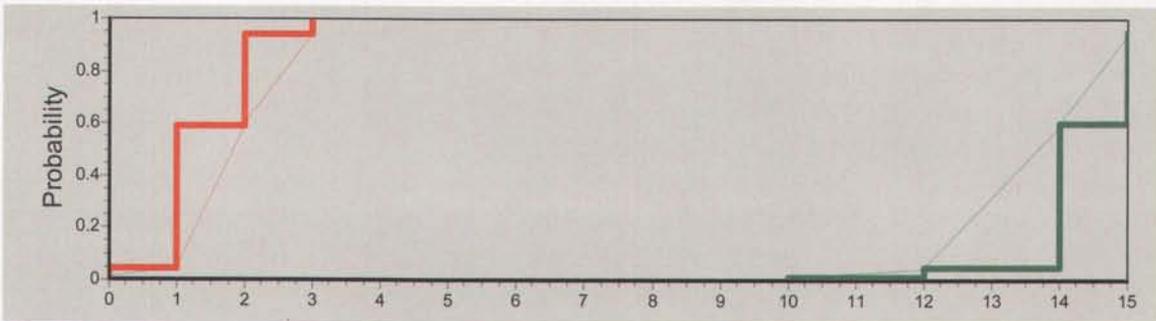


Figure 29.

The difference between these two plots reveals that, when input constraining is turned off, the p-box is constructed by *connecting the dots* between the specified percentiles. When input constraining is on, a rigorous *outward-directed rounding* method is used to connect the specified percentiles with step functions that are sure to enclose any distribution satisfying the given bounds on the percentiles.

Once you have selected exploratory mode, you cannot restore Constructor's normal behavior except by exiting the program and re-invoking it. You can exit the program by selecting File/Exit from the main menu or just pressing function key F4 while holding down the Alt key. The reason for this limitation is that, when you are exploring, the program is still accumulating histories (those hints that appear when the mouse lingers

over an input field) and keeping track of information that you don't want to be part of any actual session record. Restarting the program from scratch clears these histories. In principle you can save the inputs you have made while in exploratory mode before you exit the program, but this is not recommended. The reason is that it is possible, indeed, likely that during your exploration you will specify inputs that contradict each other.

Although you can't turn off the exploratory mode, you can turn cascading back on (by selecting Bounding/Constrained from the main menu), restore bounding to combine information from the several tabbed pages of inputs (by selecting Bounding/All pages), and restore color coding (by checking the box labeled "Color code input fields" on the options dialog which you invoke by selecting Input/Options from the main menu).

2.3 Justifications

Constructor strongly encourages you to enter explicit justifications for all the numerical entries and list choices you make in specifying what's known about the uncertain number on any of the tabbed input pages. This section explains how Constructor uses justification dialogs to let you record the reasoning and ancillary details that you had in mind when you decided the value or choice you intended to use for any of these inputs. With the justifications you enter, the program creates self-documenting input files that can help you keep track of the

- description of the value,
- original units of measurement,
- nature of the estimate or observation,
- agent or agency who estimated or vouches for the value,
- justifying arguments or reason, and
- relevant references and supporting data.

This information can be collected together in a survey by selecting Justification/Survey from the main menu. This survey can be printed or captured for inclusion in reports.

Different input fields on the several input pages have separate justifications, except that a spreadsheet (such as those appearing on the Data, Density and Graph pages) have only one justification for all the cells in the spreadsheet, considered as a group. This is mostly a matter of convenience, as the task of justifying individual data points could be quite daunting. The justification for the spreadsheet should certainly include a discussion of any outliers or other values within the data sets that merit special mention or focus.

If an input field is color-coded yellow, Constructor is telling you that the value needs justification. You needn't wait for a field to be yellow however. You can give justifications for white input fields just as well. By design, the input fields on the "Name and units" page are exempt from color coding, so they do not become yellow.

Nevertheless, you can give justifications for most of these inputs too. The only input fields for which justifications are not supported are the description and references inputs on the "Name and units" page, and the inputs comprising the justifications themselves.

Your justifications and the supporting and ancillary information for each input are organized on a justification dialog for that input. You can invoke the justification dialog by right-clicking on any input field. Right-clicking on the mean field on the Parameters page invokes the justification dialog shown in Figure 30 below.

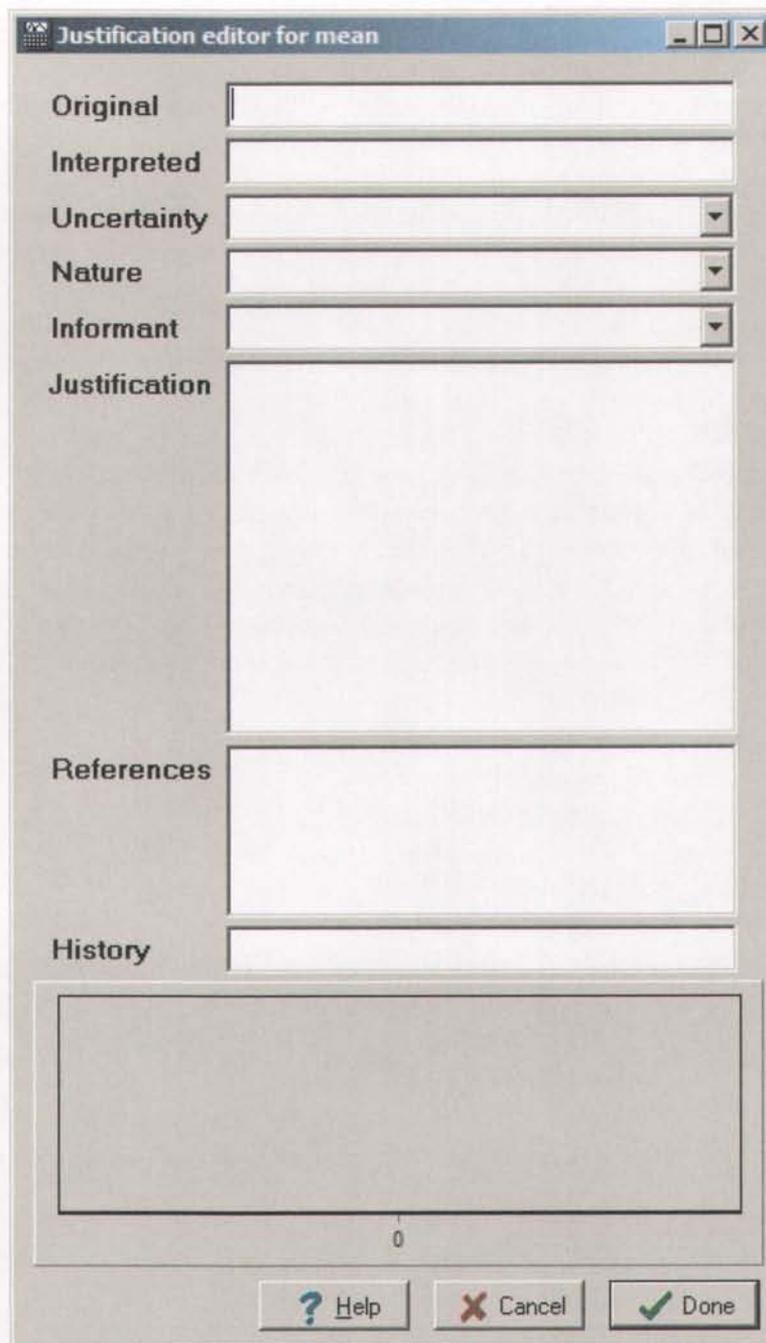


Figure 30.

This justification dialog is typical of what you'll see for any of other numerical inputs in Constructor. You can tell which input you are justifying by the title at the top of the justification dialog. Justification dialogs for list choices and non-numeric inputs have fewer fields. For instance, the ensemble field on the Name and units page only has fields for the informant, justification and references. The justification dialog for the unimodal box on the Shape page has these three plus fields for the nature and history.

Original. The first field of the justification dialog is a record of the input that you first made for the quantity. It remembers what you said you knew about the value, with its original expression of uncertainty and in its original units. The ability to recollect this original expression is obviously important for the sake of tracking information quality. If you had typed anything on the numeric field on which you right-clicked, the value would automatically appear on the original field on the justification dialog. You can edit the contents of the original field. This gives you the ability to revise and extend your original statement. Any changes you make will be reflected on the input field when you close the justification dialog.

Every numerical value representing a physical quantity (rather than some purely mathematical entity) should be accompanied by an explicit and unambiguous expression of the units in which the quantity is expressed. You may elect to have Constructor infer the units of each value you enter from the default units you specified on the “Name and units” page. If you do, then the original field can simply be a number without any units. This might be convenient when initially entering values, but it could make changing the default units later more difficult. You might want to change the default units, for instance, if you wanted to change the units of the output that Constructor produces. In any case, you can always specify the units for the quantities explicitly as you enter them. Just put the units after the numeric value, as in the expressions “[1,3] meters” or “[40 sec, 2 minutes]”. When you’re giving the units for the variance, remember that they should be squared, and that the coefficient of variation and probabilities are always dimensionless.) Generally, the units you give should be the *original* units in which the quantity was measured or otherwise estimated. If the units for any quantity are not the same as the default units you gave on the “Name and units” page, the program will automatically convert* the value to the necessary units whenever it uses the value in calculations, or generate an error message if the units cannot be so converted.

Interpreted. The interpreted field on the justification dialog spells out how Constructor interpreted your original input. This interpretation depends not only on the original input you made, but on the default units you specified on the “Name and units” page, and any subsequent inferences drawn from information about other fields.

The plot at the bottom of the justification dialog graphically displays both the original and the interpreted values. By default, the former appears in green and the latter appears in yellow. The interpreted estimate is usually only narrower than the original value’s uncertainty. The vertical distance on the plot is not meaningful.

Uncertainty. The uncertainty field on the justification dialog provides a space for you to characterize the uncertainty present in your expression for the quantity. You can type anything at all you like in this field. If you click on the little down arrow at the far right of the input field, you’ll see a list of the kinds of characterizations you might make. If you do, the display will look like Figure 31.

*Unless you’ve unchecked the “Automatically convert” checkbox in the lower, right-hand corner of the Input/Options dialog, in which case all the units you enter must already conform.

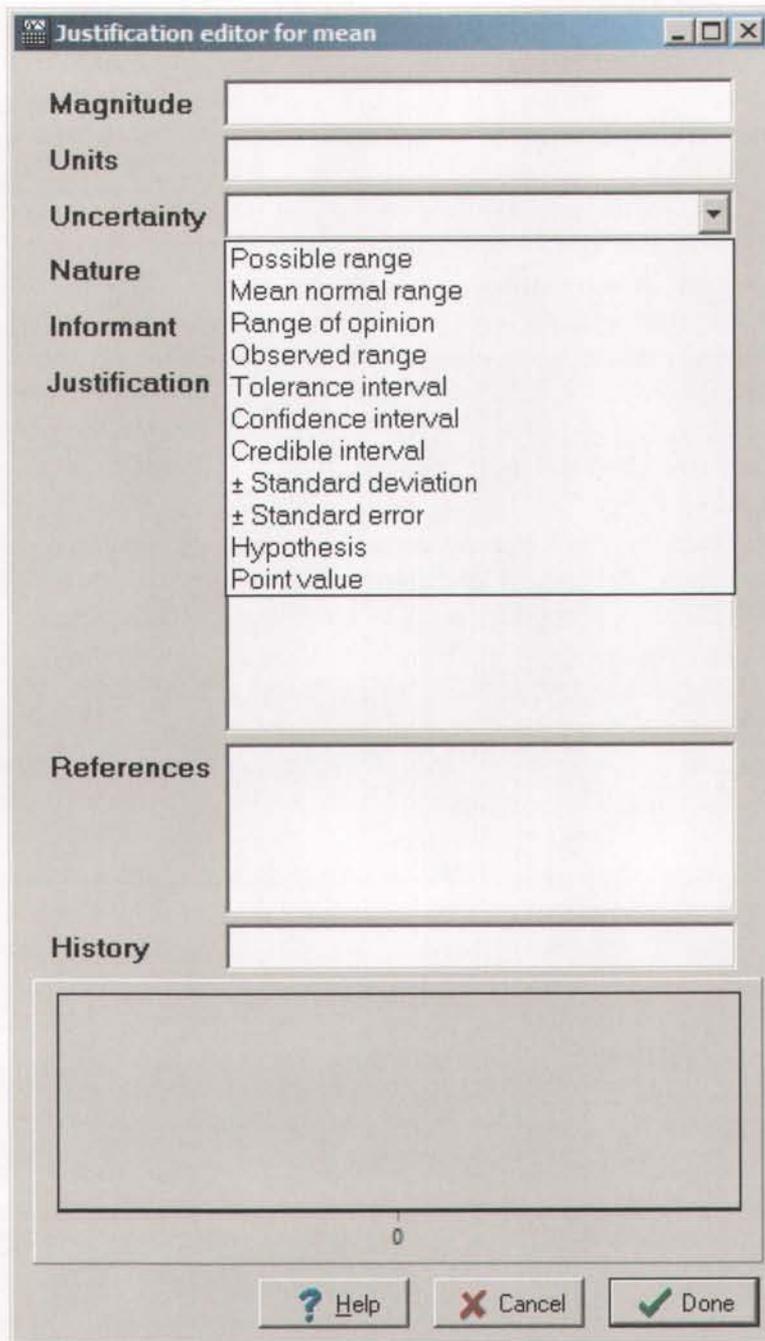


Figure 31.

You can choose one of these characterizations of uncertainty by clicking on it in the drop down list. Doing so will overwrite any text you might already have entered in the field. If none of these characterization quite fit, don't hesitate to give a different characterization that is more appropriate for your situation and approach.

Nature. This field provides a space for you to characterize the nature of the value and the circumstance under which the value you've used for the quantity was estimated. Clicking

the down arrow at the right of the field invokes another drop down list of choices. The resulting display will look like Figure 32.

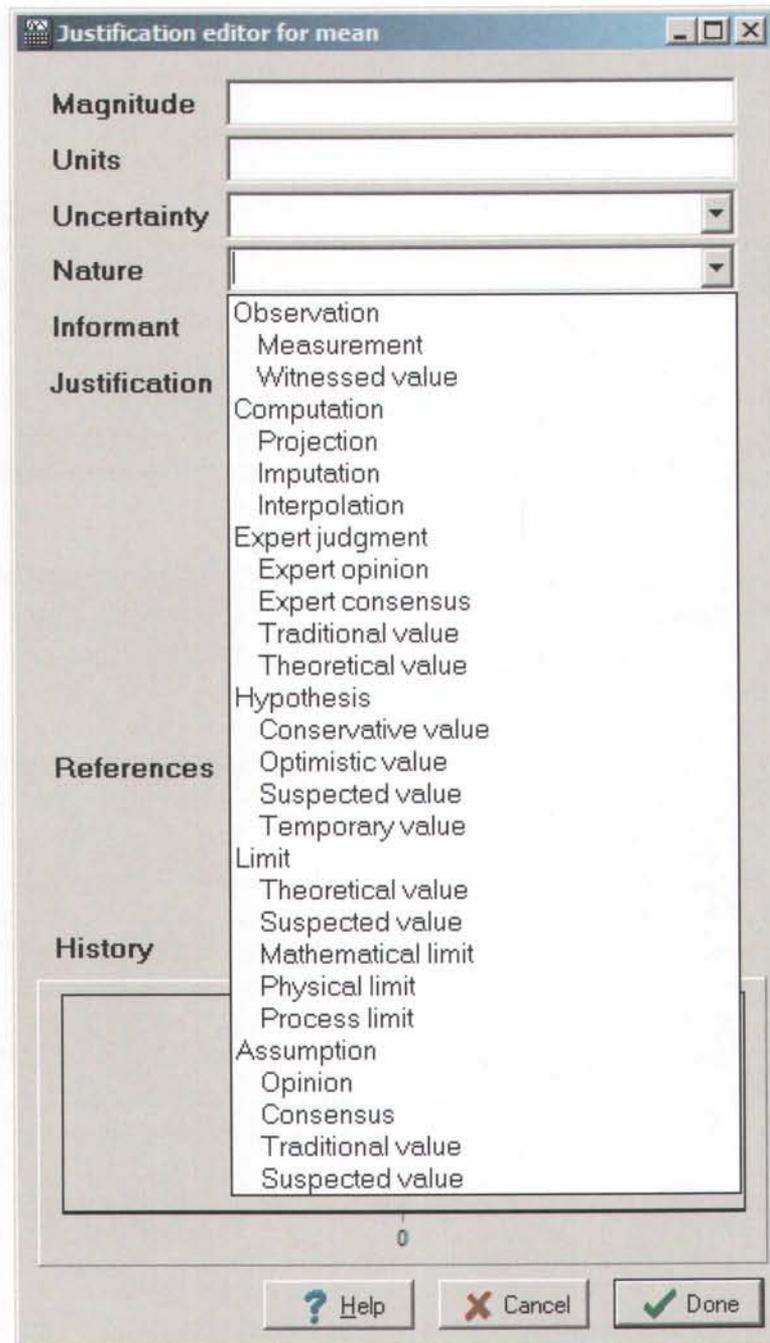
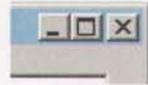


Figure 32.

This is a convenient list of characterizations, one of which may be appropriate to describe the nature and origin of the parameter. You can select any of these choices (including the headers) or type in your own characterization if none of these is fully appropriate for a particular parameter. The idea is to say just what the value you've given for the parameter really represents. If it's a guess, you should say so.

Informant. This field lets you name the person or institution on whom you relied for the information about the quantity. The down arrow invokes a drop down list of possible informants from which you could select. You can edit the contents of the drop down list by selecting Justification/Informants from the main menu.

Justification. This multiline field is for entering the justification and any supporting arguments for the parameter value you entered. You can type whatever you like for the justification. You can use bold face, italics, underlining, colors, and alternate fonts to emphasize or structure your text (see the section “2.4.1 Editing and formatting text entries” on page 50 for hints). The text you enter on these fields is not limited. If you enter more than can fit into the field, the field will allow you to scroll. You can maximize the display window by clicking on the box immediately to the left of the little × which appears at the top, right-hand corner (shown at right) of the justification dialog. You should enter as much detail as you might need to later recount your thinking that led to the input that you made for the parameter in question.



References. Place references to the scientific or engineering literature in this field that support or to which you refer in the justification. References can include personal communications and arbitrary sources of information. Boldface, italics, underling, text color and font are under your control.

History. This field is contains a record of how the estimate for the quantity has been updated. Users are not expected to edit this field, although they can if they wish to.

The dialog in Figure 33 is an example of what a minimally completed justification might look like for the parameter mean. Of course, your standards for what is complete may differ. Generally, one can expect that conscientious effort in documenting one’s inputs will be rewarded.

The graph at the bottom of the justification dialog plots both the original input (in green) and its interpreted value (in yellow) if it is possible to do so. The heights of the graphs are meaningless, and they are offset so that you can easily distinguish them. These graphs provide a check that Constructor is understanding the inputs that you are making.

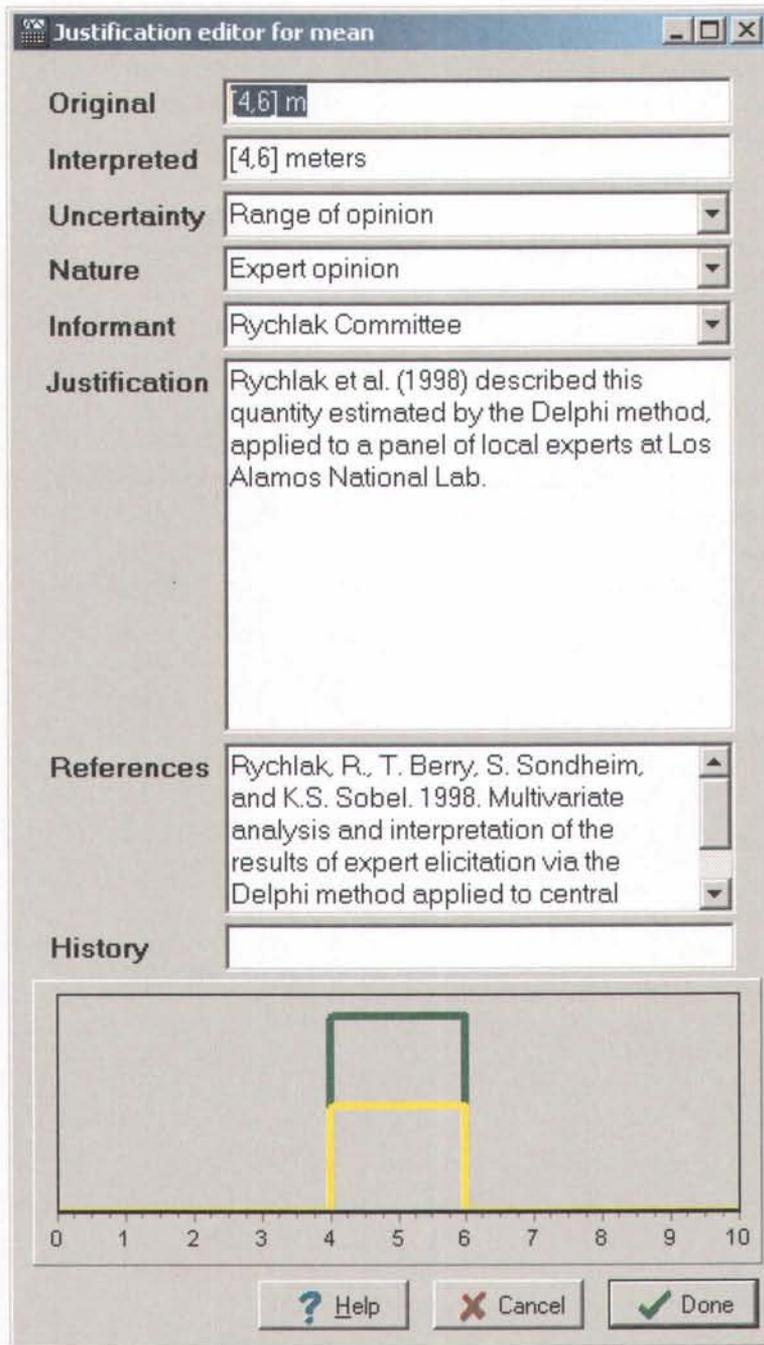


Figure 33.

The justification dialog can be closed by clicking the Done button. If color coding for input fields is turned on, the yellow input field for the mean parameter would then be shown in white. You can select Input/Options from the main menu to change the colors used by this system, or to turn it off altogether.

2.4 Input and editing

This section documents input conventions used by Constructor and reviews user guidance for efficiently entering data and program selections. It covers how inputs are made, changed and erased from numeric and text fields and spreadsheets. See section 5 on page 78 for details on choosing program options.

2.4.1 Editing and formatting text entries

When you type characters in a text entry field, they appear at the current position of the text cursor. The position of the text cursor is indicated by a slowly blinking vertical bar. You can move the text cursor by pressing the arrow keys (Left arrow, Right arrow, Up arrow, Down arrow) or by pointing to a spot with the mouse and pressing the left mouse button, i.e., by clicking where you want the text cursor to be.

Constructor supports most of the standard keystroke conventions for editing and navigating that are common in Windows applications. For instance, the following keystrokes perform the associated functions.

<i>Keystroke</i>	<i>Function</i>
Control-C	Copy selected text into the clipboard
Control-V	Paste text from the clipboard
Control-X	Cut selected text to the clipboard (i.e., copy and then delete it)
Control-Z	Undo the last edit
Delete	Erase the character to the right of the text cursor
Backspace	Erase the character to the left of the text cursor
Left arrow	Move the text cursor one character to the left
Right arrow	Move the text cursor one character to the right
Home	Move the text cursor to the start of the line
End	Move the text cursor to the end of the line
Control-Left	Move the text cursor one word to the left
Control-Right arrow	Move the text cursor one word to the right
Shift-Left arrow	Select (or deselect) a character to the left of the text cursor
Shift-Right arrow	Select (or deselect) a character to the right of the text cursor
Shift-Home	Select (or deselect) all text from the cursor to the start of the line
Shift-End	Select (or deselect) all text from the cursor to the end of the line
Double click	Select a word
Tab	Move to the next input field or control
Backtab	Move to the previous input field or control

The text fields that allow for more than a single line of input also support a few other keystrokes, including some to toggle various formatting options that let you emphasize or highlight text in different ways. The following additional keystroke conventions are supported for multi-line text fields.

<i>Keystroke</i>	<i>Function</i>
Control-B	Toggle boldface formatting
Control-I	Toggle italic formatting
Control-U	Toggle underline formatting
Control-F	Invoke a font dialog on which you can change the font
Insert	Toggle insert/overtyping mode

Enter	Insert a carriage return at the text cursor
Up arrow	Move the text cursor up one line
Down arrow	Move the text cursor down one line
Page Down	Move the text cursor to the beginning of the entry field
Page Up	Move the text cursor to the end of the entry field
Shift-Up arrow	Select (or deselect) all text from text cursor to the previous line
Shift-Down arrow	Select (or deselect) all text from text cursor to the next line
Shift-Page Up	Select (or deselect) all text from text cursor to the start of the field
Shift Page Down	Select (or deselect) all text from text cursor to the end of the field

Selecting text. You can use the shifted arrow keys to select text. You can also select text with the mouse. You can double click on a word to select it. You can also select text by clicking at one end and, while holding down the mouse button, repositioning the mouse to the other end of the text to be selected. You can deselect text by pressing any of the non-shifted arrow keys or clicking the mouse anywhere in the field.

Moving text. You can move text that you have already entered by selecting it, pressing Control-X, moving the text cursor to where you want the text to be, and pressing Control-V. The multi-line text fields also support dragging and dropping text. To drag and drop text, first select it and then click on the selection (but do not release the mouse button). While holding the mouse button down, reposition the mouse to the spot where you want the text to be and then release the button to move it there.

Deleting text. You can delete text one character at a time with the Backspace or Delete keys. You can also delete text by selecting it and pressing the Backspace, Delete or Control-X key. Pressing the Control-V key replaces any selected text with what is in the clipboard.

Copying text. You can copy text by selecting the text and pressing the Control-C key. Thereafter, whenever you press the Control-V key the copied text will be inserted at the current position of the text cursor. The clipboard is overwritten each time you press the Control-C or Control-X key when some text is selected.

Changing format or font. (You cannot change the format or font of the single-line text fields.) Press the Control-F key to bring up a little font dialog. You can change the font face, size, style, and color with this dialog. Press the OK button after you've made your choices. If there was text selected when you pressed the Control-F key, your choices will be applied to the selected text. If no text was selected when you pressed the Control-F key, the choices affects all text you subsequently type until you move the text cursor with the mouse or the arrow keys or you otherwise alter the format or font. The keys Control-B, Control-I and Control-U are shortcuts that toggle on (or off) boldfacing, italics and underlining respectively, or likewise modify the format of any selected text. Copied or moved text retains it formats and font. Any new text that you enter will appear in the format and font of the text on the left of the text cursor.

2.4.2 Entering and editing data in spreadsheets

Constructor has spreadsheets on the Data, Density and Graph pages. In the latter two pages, the spreadsheets become visible by double-clicking on the graphs.

You can paste numbers into a Constructor spreadsheet that you have previously copied into the Windows clipboard from another program. Click on the upper, leftmost cell where you want the number to be placed and press Control-V or select Edit/Paste from the main menu. The contents of the clipboard will be placed in the spreadsheet at the cell on which you clicked. You can paste a single number into a one cell, or a (rectangular) range of numbers into adjacent cells all at once. You can paste multiple columns at once, but only the first three columns on the clipboard will be copied. Up to 1000 rows can be copied. If you had more data in the clipboard than could be read into Constructor's spreadsheet, no warning message will be given, so it is always prudent to check that all the data arrived safely where you intended.

You can also copy cells from the spreadsheet to the clipboard. Click on a corner cell of the range of cells you want to copy and highlight the range of cells by pressing the arrow keys while holding down the shift key. (Constructor doesn't support the use of the mouse to select cell ranges.) When the range of cells you want to copy is highlighted, press Control-C or select Edit/Copy from the main menu to copy the cells to the clipboard.

The cut, restore, and clear functions on the Edit on the main menu have no effect on spreadsheets. Pressing Control-X in a spreadsheet will only cut selected text within a single cell, not ranges of cells. Selecting Edit/Reset from the main menu will erase *all the cells in the spreadsheet*, rather than merely the current cell or a range of selected cells. It asks you for a confirmation before resetting the entry and its justification, so beware if you get such a question.

In practice, you will probably use the paste function to enter numbers into the spreadsheet, but Constructor also allows you to enter data directly in a spreadsheet. To do so, you must first make the cell you wish to edit the current cell, either by clicking on it or by using the tab, back-tab, or arrow keys to navigate to it. You can then type the appropriate value. Move to the next cell by again pressing tab or one of the arrow keys.

Before you click on it or tab into it, the spreadsheet does have the program's focus. To make edits to the spreadsheet, you must first give it the focus by clicking in the cell you want to edit or tabbing into it. Once you do, the current cell in the spreadsheet will contain the typing cursor (a slowly blinking vertical line) and may show text in reverse highlighting. If the blinking cursor is not visible, or if an entire cell of the spreadsheet is highlighted (either a dark rectangular cell, possibly with white letters or a white cell on a yellow spreadsheet), rather than merely the text within the cell, the spreadsheet has lost the program focus, and you may need to click on it or tab into it again.

Constructor asks for only one justification for a spreadsheet. The justification you supply should refer to the entire data set it contains. To enter the justification, right-click anywhere on the spreadsheet and the justification dialog will appear.

2.4.3 Clearing and resetting entries

They call it *research* because you have to keep looking, over and over again, for the right answer. Inevitable mistakes are part of the deal. Constructor supports several operations to undo inputs that you've made. In increasing order of comprehensiveness, these are:

<i>Access</i>	<i>Operation</i>
Backspace	Deletes the last character you typed,
Control-Z	Restores an entry to its state before you clicked on or tabbed into it,
Edit/Restore field	Restores an entry to its state before you clicked on or tabbed into it,
Edit/Clear field	Clears an entry,
Edit/Reset field	Erases the entry and its justification,
Reset button	Erases all entries and their justifications on the current tabbed page,
File/New	Erases all entries and their justifications on all pages.

The restore and clear functions apply to entire fields. They work on multi-line input fields and listboxes, but they cannot be applied to an entire spreadsheet or to graphical edits. You have to undo changes in spreadsheets one cell at a time, and in graphs one point at a time. The reset functions erase not only the entry itself but any justification it may have, including automatically generated histories. For this reason, you have to answer a confirmation request before the entry is actually reset. You can reset textual entries, listbox entries, and *entire* spreadsheets. To reset one of the graphs in the lower part of the display, double-click on the graph to bring up the corresponding spreadsheet or spreadsheets, and then reset the spreadsheet(s). The graph in the upper part of the display is always recomputed from the available information, so you don't need to reset it.

Keep in mind that none of these various functions will undo any of the automatic cascades to other fields that might have been induced by your input. If you're having trouble with Constructor prematurely applying the automatic cascading to inputs that you consider only tentative, it might be best to start over and click the Relax button, which turns off automatic cascading. See page 81 for more information about automatic cascading of inputs.

2.5 Units

Constructor recognizes and propagates the units of the inputs you make.

You can enter the units you want to use for Constructor's output on the "Name and units" page. Remember that you can express the exponents for units in curly braces or with the words "square" or "cubic". The words "per" and "over" are also understood. For example, the units "square meters", " m^2 ", "meter²" and even "m meters" are all the same. Likewise, the expressions " $kg\ m\ s^{-2}$ ", "kilogram meter per square second", " $kg\ s^{-2}\ m$ ", " $kg\ m\ over\ s^2$ ", "kilogram per second per second meter", "Newton", "N" are all equivalent.

You can ask Constructor to assume your inputs are appropriate given the default units (which you specified on the "Name and units" page) for all the inputs you make anywhere in the program. This feature is offered as a convenience for the user, but it has disadvantages. To imagine what its disadvantages are, recall the unhappy conclusion of NASA's Mars Climate Orbiter mission (Isbell et al. 1999). If you use this feature, the program will assume that any number you enter in a field is appropriate for that field and

does not require any automatic units conversion. Thus, if the default units are “meters”, then an entry for the mean would be presumed to be in meters, and an entry for the variance would be presumed to be in square meters. Entries for the coefficient of variation and probabilities would be assumed to be unitless. (Probabilities appear as the rightmost values on the Coverages page, as the densities on the Density page, and as a parameter for a few of the named distribution families on the Shape page.)

If can always enter the correct units together with the numerical value on the input field. Your entry might be something like “[1.2, 1.5] meters”, but it could also be “[120 cm, 150 cm]” or even “[3.94 ft, 1.5 m]”. Constructor will parse and interpret these entries. If they require unit conversion to be compatible with the default units, the software will carry out the conversions automatically. As a matter of good analytical practice, you should generally enter a quantity in the original units in which it was measured or assessed.

If you enter a value with units that do not conform appropriately with the default units, Constructor will complain with an error message. For instance, suppose the default units are meters. If you enter “12 seconds” for any input, or if you enter “12 meters” for the variance, Constructor will slap your wrists.

2.6 Programmatic limits and specifications

The textual inputs you make in Constructor are generally unlimited except by your computer’s resources. There are, however, certain programmatic limits on some of Constructor’s capacities:

1×10^{100}	Largest numeric value
-1×10^{100}	Most negative numeric value
1×10^{-100}	Smallest absolute value
16	Maximum number of decimal digits in numeric output
1000	Maximum number of data intervals
0, 80, ..., 99, 100	Possible confidence levels for Kolmogorov-Smirnov bounds
10	Maximum number of coverage statements
1000	Maximum number of points in a bubble
1000	Maximum number of points in a cap
40	Maximum line thickness for bounds in graphs
25	Maximum length in characters of an individual unit
100	Maximum length in characters of the entire unit string
999	Maximum number of half-seconds for duration of hints
105	Number of explicit probability levels in the discretization (at 0, 0.0001, 0.001, 0.01, 0.02, ..., 0.98, 0.99, 0.999, 0.9999, 1)

In most cases, these limits and specifications are arbitrary. If you need to change them for an application you would like to make of Constructor, or, indeed, if you would like to suggest other changes or enlargements of functionality or improvements of its design, the software developers invite you to contact them (see “Technical support” on page 5) about your ideas.

3. Output

This section describes how Constructor's output is configured and how you can format it to suit your needs. The first section below is a survey of the various issues controlling the nature and form of the outputs. The following section is the second of the two program tutorials in this document. The tutorial walks through the steps of specifying the outputs you want. After the tutorial, there is a brief review on page 62 of the use of the maximum entropy criterion for selecting precise distributions when the available information only justifies a class of possible distributions. Following that, on page 65, is a synopsis of the programmatic conventions for rescaling and capturing graphs.

3.1 Synthesizing constraints to select distributions

Constructor allows you a great deal of flexibility about the output it produces. There are five distinct issues under your control: whether all or only some information should be used, the aggregation method, the nature of the output, the pinching criterion, and the output format.

All or some information. This choice determines whether the characterization of the uncertain number should be based only the information you specified on Constructor's current page of inputs, or whether it should be based on all the information you specified on any of the tabbed pages ('Shape', 'Parameters', 'Data', etc.) combined together. The default is that the information from the all the input pages is combined, and, in practice, one would presumably want to use all available information to characterize an uncertain number. This default may sometimes be cumbersome however, especially when you are beginning a new analysis of a quantity. You can change the behavior by selecting Bounding/Current page from the main menu. The output would then be based only on whatever information is specified on the current page of inputs (or the last page of inputs selected). You can restore the default behavior by selecting Bounding/All pages on the main menu.

Aggregation method. If all the input information from the various tabbed pages ('Shape', 'Parameters', 'Data', etc.) is to be combined, you can also choose how the information should be aggregated. You may want to intersect the constraints formed from the information on each of the pages, or you may want to envelope them. The default is to use intersection as the aggregation method. This means that information specified on a page can only tighten the resulting p-box or Dempster-Shafer structure. This is reasonable if all of the information is true. The disparate information on the various pages must of course be consistent. If, however, you are unsure which set of information is accurate, it may be reasonable to use enveloping as the aggregation method by selecting Bounding/Envelope from the main menu. This choice means that specifying disparate information on different pages can widen the resulting uncertain numbers. You can restore the default aggregation method by selecting Bounding/Intersection from the main menu. If the information is not combined from the various tabbed pages of inputs because Bounding/Current page has been selected as described above, then no aggregation method is used and the Bounding/Intersection and Bounding/Envelope options from the main menu are unavailable.

Nature of the output. Constructor produces three kinds of outputs that characterize an uncertain number. It can give you a precise probability distribution, a p-box, and a Dempster-Shafer structure. By default, Constructor outputs both a probability distribution and a p-box, but you may elect to have the software output any of the three, or any two, or all three. You can choose which outputs you want by checking or unchecking the three corresponding boxes along the top of the options dialog, which is accessed by selecting Input/Options from the main menu. When you subsequently export an uncertain number, Constructor will ask you for a file name for each kind of output you've asked for.

Pinching method. If a probability distribution is desired but the specified information has not narrowed the choice down to a precise one, then we have to use some additional pinching criterion that will select a particular distribution from the class of distributions that would be possible under the given constraints. Constructor offers seven possible choices for the pinching method. The choices are on the far left side of the options dialog which you can access by selecting Input/Options from the main menu. The default is vertical averaging, which estimates the probability distribution as the vertical average of its upper and lower bounds. The choice about pinching is only relevant if you want a precise probability distribution as the output. If only a p-box or a Dempster-Shafer structure is the output, then no pinching method is needed or used. The tutorial in the next section considers pinching in some detail.

Output format. You can also select the format of the output. Many formats are available, including a graph displayed on Constructor's display, a graph saved in metagraphics format to a disk file, numerical ASCII text files in various configurations, and files specially configured for reading by other programs. A graphical display is always created automatically. It can be saved in metagraphics format by right-clicking on the graph and selecting the copy graph option from the popup menu. Clicking on the Export button creates a file or files for a probability distribution, p-box and Dempster-Shafer structure formatted according to settings on the dialog you see when you select Input/Options from the main menu.

3.2 Program tutorial: getting output

This tutorial will demonstrate how to select the nature of the output, the pinching method and the output format. Invoke Constructor and select File/Open from the main menu. Enter the file name OUTPUT.CON in the Open dialog, and click on its Open button to read the file into Constructor. Click the Data tab to open the Data page. Constructor can make three kinds of outputs and we will create them all. Select Input/Options from the main menu. Then click on the little white box labeled "Dempster-Shafer" that appears in the middle of the top of the options dialog so that the box is checked. The box labeled "Probability distribution" in the upper, left-hand corner and the box to its right labeled "Probability box" should already be checked. (If they're not, click on them to check them.) You can close the options dialog by clicking on the Done button in its lower, right corner.

You are now ready to generate some output. You can do this by clicking on the Export button at the bottom of the main display, or, equivalently, by selecting File/Export from the main menu. This will bring up a series of three filename dialogs, one for each

of the three outputs. Notice that at the top of the first dialog, the title is “File for the probability distribution”. Enter the file name “export” in the “File name” field, and click the Save button. The probability distribution that is written to the file is depicted in red on the graph on the upper part of Constructor as shown in Figure 34. By default, the file extension for probability distributions is .CDF, so the output will be written to the file EXPORT.CDF.

Another filename dialog will immediately appear with the title “File for the p-box”. It remembers the name you just gave, so you can just click the Save button or press the Enter key. The default extension for p-boxes is .PB, so the output file created will have the name EXPORT.PB. Finally, a third filename dialog with the title “File for the Dempster-Shafer structure” will appear. The default extension for Dempster-Shafer outputs is .DS, so pressing the Enter key again will send the Dempster-Shafer output to the file EXPORT.DS. For both the p-box and the Dempster-Shafer structure, the sequences of points in the output are depicted as the red and white bounding distributions on Constructor’s upper graph, as shown in Figure 35.

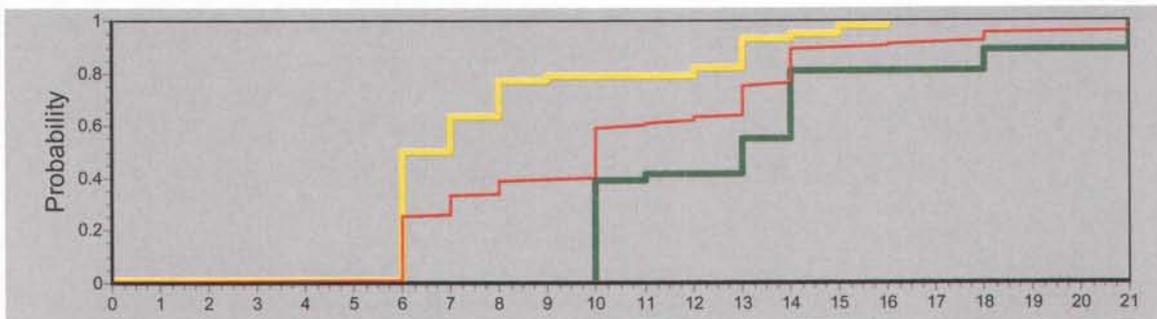


Figure 34.

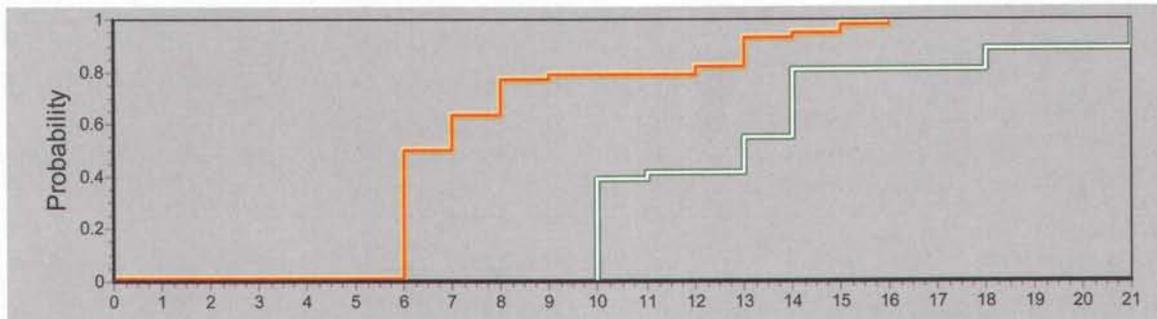


Figure 35.

You can change the colors and the line thickness used to display the output with the settings in the Export frame on the options dialog you access by selecting Input/Options from the main menu.

You may have noticed that, even over ranges of x where the upper and lower bounds are both horizontal, there seems to be a slightly positive slope in the red exported distribution. Given that the pinching is vertical averaging, this should be surprising. It is the result of discretization used to make the calculations. Although the upper and lower bounds are expressed with separate sequences of points and outward-directed rounding, the exported distribution is specified with only a single sequence of points. The red curve

connects the dots between these points and this is what induces the slight upward slant in places where one would otherwise expect a horizontal shoulder in the distribution. This is offered as the best single distribution, discretized in 100 steps, to represent the uncertain number.

You can use any text editor such as Notepad or Wordpad to examine the files created by Constructor. If you do so, the files should look like the following.

EXPORT.CDF

```
; Risk Calc version 4.0
; Units:
; Type: RANDOM
0      0.000000000000000E+0000  0.000000000000000E+0000
1      0.000000000000000E+0000  6.000000000000000E+0000
2      6.000000000000000E+0000  6.000000000000000E+0000
3      6.000000000000000E+0000  6.000000000000000E+0000
:
:
```

EXPORT.PB

```
; Risk Calc version 4.0
; Units:
; Type: RANDOM
0      0.000000000000000E+0000  1.000000000000000E+0001
1      6.000000000000000E+0000  1.000000000000000E+0001
2      6.000000000000000E+0000  1.000000000000000E+0001
3      6.000000000000000E+0000  1.000000000000000E+0001
:
:
```

EXPORT.DS

```
0.000000000000000E+0000  1.000000000000000E+0001  0.01000
6.000000000000000E+0000  1.000000000000000E+0001  0.38000
6.000000000000000E+0000  1.100000000000000E+0001  0.03000
6.000000000000000E+0000  1.300000000000000E+0001  0.08000
7.000000000000000E+0000  1.300000000000000E+0001  0.05000
7.000000000000000E+0000  1.400000000000000E+0001  0.09000
8.000000000000000E+0000  1.400000000000000E+0001  0.13000
9.000000000000000E+0000  1.400000000000000E+0001  0.02000
1.200000000000000E+0001  1.400000000000000E+0001  0.02000
1.200000000000000E+0001  1.800000000000000E+0001  0.01000
1.300000000000000E+0001  1.800000000000000E+0001  0.07000
1.300000000000000E+0001  2.100000000000000E+0001  0.04000
1.400000000000000E+0001  2.100000000000000E+0001  0.02000
1.500000000000000E+0001  2.100000000000000E+0001  0.03000
1.600000000000000E+0001  2.100000000000000E+0001  0.02000
```

The default formats for probability distributions and p-boxes is a format that can be read by RAMAS Risk Calc's (Ferson 2002) import function. These files are ordinary text so they can also be easily read into Excel or other post-processing software, but you might prefer a different format if this is your intended use. You can explore the various output formats that Constructor supports by selecting Input/Options from the main menu and inspecting the various radio buttons associated with the three kinds of output types.

They are assembled together on the dialog within frames under the three respective check boxes.

In the case of the Dempster-Shafer output, the first two columns define the interval of the focal element, and the third column is the probability mass associated with each focal element. Note that, unlike the files for the probability distribution and the p-box which contain over one hundred lines, the file for the Dempster-Shafer structure has only fifteen lines. Because the default number of masses on the options dialog for this structure is “condensed”, the output is fashioned by horizontally slicing the p-box representing the uncertain number. Fifteen is the number of horizontal slices that best captures the Dempster-Shafer structure derived from the ten weighted input intervals displayed on the spreadsheet on the Data input page. You could also have requested the number of masses simply be determined by the original data by checking the appropriate radio button on the options dialog. This would produce an output file with ten lines looking very much like the original data except that the weights would be scaled so they sum to unity.

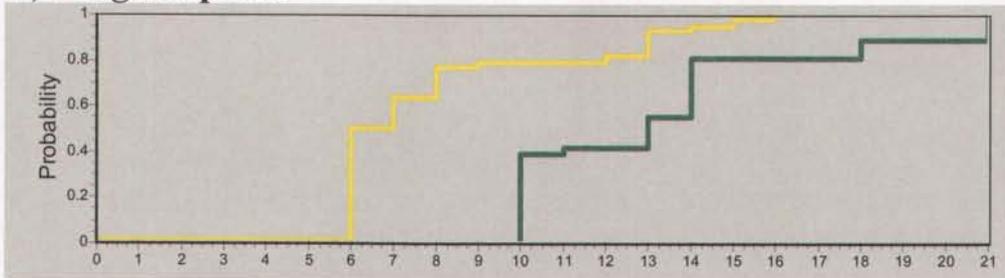
3.2.1 Pinching

The suggestion that the probability distribution depicted as a red curve is the “best” single distribution to represent the uncertain number is not without qualification. There are actually many possible precise distribution functions that could be chosen that would be consistent with the uncertain number. Constructor supports several alternative pinching methods that can be used to make this choice. The rest of this tutorial will explore the effect of the pinching methods. See Sentz and Ferson (2002), Ferson et al. (2003) and Abbas (2003) for descriptions of the mathematical methods and algorithms employed to compute these distributions.

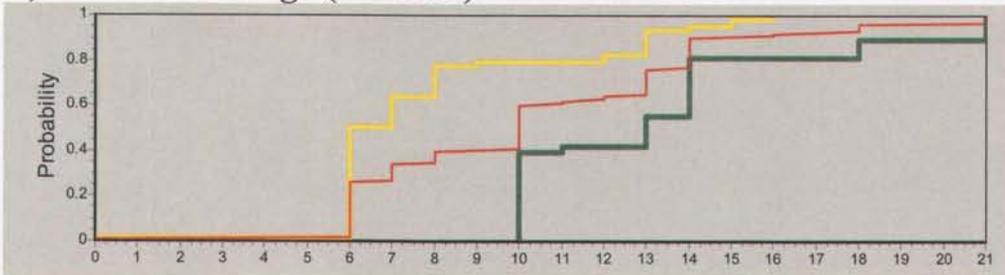
Select Input/Options from the main menu. Click on the checkboxes for the probability box and the Dempster-Shafer structure so that neither is checked. Only the leftmost checkbox for the probability distribution should be checked. This time, you won't close the options dialog, but just move it to the right of your computer screen so that only the far left column of the dialog under the probability distribution checkbox is visible. You can move the dialog by clicking on the bar at the top of the dialog and, while holding down the mouse button, repositioning the mouse so the dialog also moves. You may also want to reposition the main Constructor display so that it and the far left column of the options dialog are both visible at the same time. On the options dialog, check the ratio button labeled “Spanning” which is the second choice in the Pinching group of radio buttons. Click on the Export button again, and click the Save button on the file dialog or just press the Enter key to let Constructor write the output to the same file. The exported probability distribution is depicted as the red curve in the graph as shown in graph C of Figure 36.

The sequence of graphs in Figure 36 and Figure 37 depicts results of applying the seven available pinching methods for choosing a particular probability distribution from within a probability box. You can recreate these results by selecting in turn each of the pinching methods on the options dialog and clicking the Export button at the bottom of the main display. (For this tutorial, you only want to visualize the effect of pinching and don't need to actually create an output file. You can bypass the filename dialog by clicking on the Export button while pressing down on a shift key.)

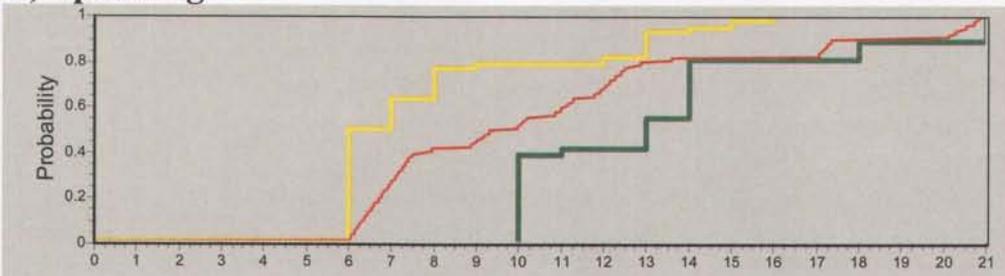
A) Original p-box



B) Vertical average (mixture)



C) Spanning



D) Dispersive (largest variance)

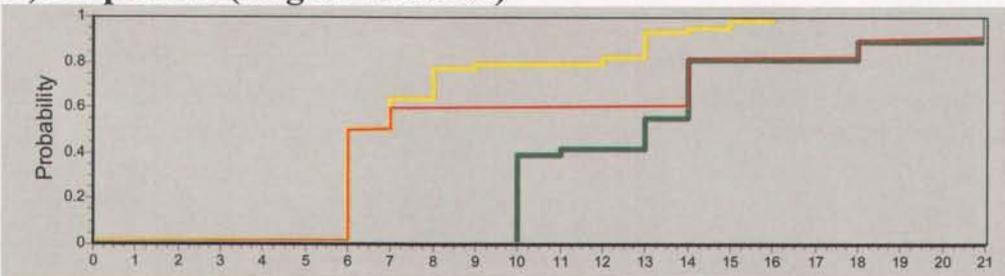
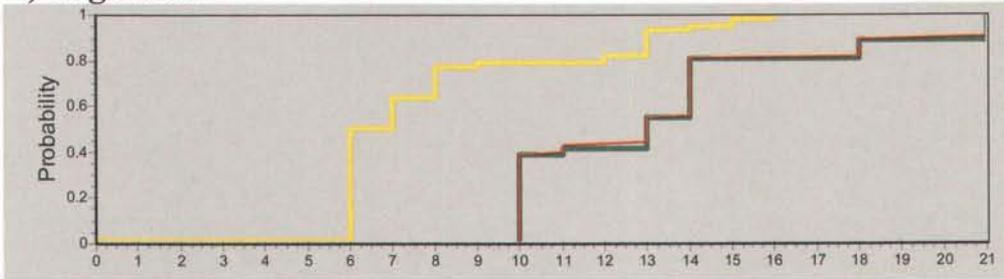
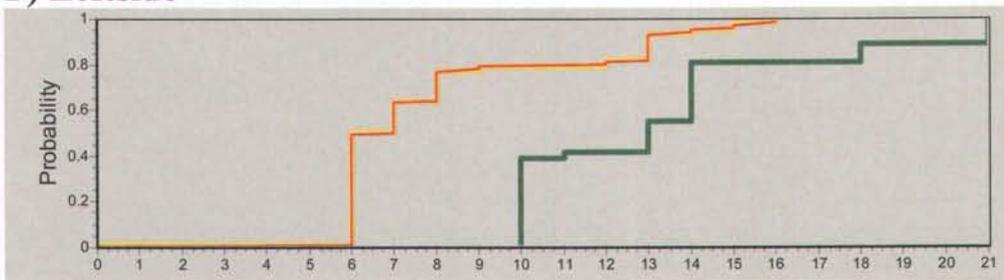


Figure 36. A p-box (yellow and green) and three different precise probability distributions (white) selected under different criteria.

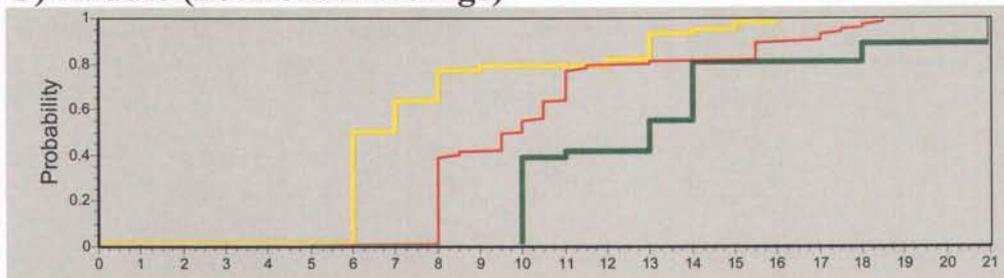
E) Rightside



F) Leftside



G) Middle (horizontal average)



H) Maximum entropy (taut string)

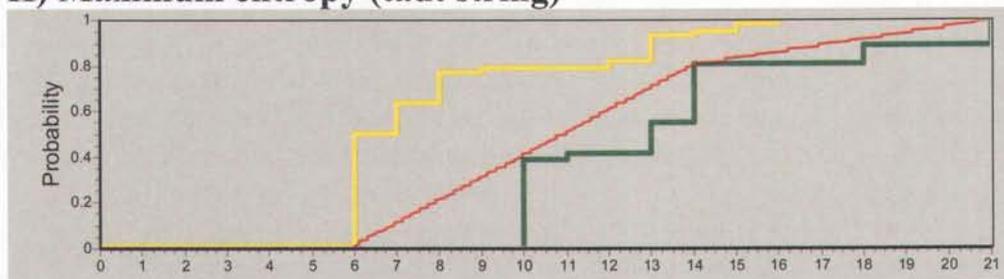


Figure 37. Four more different precise probability distributions (white) selected under different criteria from within a p-box (yellow and green).

The topmost graph A in Figure 36 shows the original p-box in yellow and green computed as the stochastic mixture of the data in the input file OUTPUT.CON. The graph B below it shows this p-box again together with the precise distribution derived from it based on the vertical average of the upper and lower bounds of the p-box. This distribution is the same as a stochastic mixture of the upper and lower bounds of the p-box. The third graph C in Figure 36 shows the spanning distribution, which is obtained as a horizontally averaging the left and right bounds with weights that favor the left bound for small values of probability and favor the right bound for large values of probability. The resulting distribution spans the range of the p-box. The bottom graph D shows the dispersive distribution, which is simply the distribution with the largest variance that can be inscribed within the p-box. It is obtained by following the left bound for small values of probability and then jumping to the right bound for large values of probability. The jumping point is chosen so as to maximize the variance of the resulting distribution.

The top graph E of Figure 37 depicts the precise distribution corresponding simply to the right side of the p-box. The next graph F likewise shows the distribution corresponding to the left side of the p-box. These choices could be appropriate if you wanted to be sure to be conservative about the selection of the distribution in the face of your uncertainty about it (and you can tell in which direction exceedances would be most adverse). The third graph G in Figure 37 depicts a precise “middle” distribution chosen by horizontally averaging the left and right bounds of the p-box. Note that this horizontal average is decidedly different from the vertical average in graph B of Figure 36. The last graph H in Figure 37 depicts a precise distribution selected according to the maximum entropy criterion based on constraints on the percentiles (i.e., a p-box). The algorithm employed to compute this result yields Abbas’ (2003) “taut string” distribution whose shape follows the shortest path connecting the corners of the p-box that is constrained to lie inside the p-box.

All of the seven precise distributions obtained by pinching are substantially different from one another in various ways. Some of the choices are averages of various kinds, and some are extremizations of something (variance, entropy, magnitude). Some emphasize one or both distribution tails, and some ignore the tails. Obviously, these different choices will have different uses in practice. There can be very little general guidance about which criterion to use to pinch a p-box to a probability distribution, other than the most general advice not to do it at all. Pinching is only ever necessary if you insist that your output be given as a single, precise probability distribution.

Pinching under any criterion necessarily ignores admitted uncertainty about the distribution. It is a way to select a precise probability distribution from a class of distributions that are equally justifiable given the inputs that you have specified. Pinching is not a method of aggregation comparable to intersection or enveloping. Instead, it is a method of *erasing uncertainty*. It may often be much better analytical practice to decline to pinch away this uncertainty, and to summarize the output from Constructor as a p-box or a Dempster-Shafer structure. When the uncertainty about the distribution is small, this will become clear as the upper and lower bounds on the p-box come closer and closer together, or equivalently, as the focal elements of the Dempster-Shafer structure become smaller and smaller intervals.

3.3 Maximum entropy

The maximum entropy criterion is a pinching method by which a single probability distribution having largest statistical entropy is identified among all the distributions consistent with available information. Figure 38 depicts nine examples of the use of the maximum entropy criterion to select a single distribution to represent an entire class of probability distributions. The available information that defines the class is indicated by the words labeling each of the nine graphs. Each graph shows the p-box in blue associated with the class of distributions, and the maximum entropy distribution in red. For instance, the class represented in the upper, left-hand graph is defined as all distributions having a given minimum and a given maximum. In this illustration, the minimum happens to be 3 and the maximum happens to be 10, but examples with other values would look very similar so that the p-box would always look like a rectangle and the maximum entropy distribution would always look like a straight line. The middle graph on the top row describes a class of distributions having a given mean and a given standard deviation. In this case, the mean is 5 and the standard deviation is 1. The rightmost graph on the top row describes a class of distributions each having values that are strictly positive and a 75th percentile equal to 6. The blue p-box is truncated at 50, but in theory goes to positive infinity.

In Constructor, the “taut string” distribution that connects the far corners of a p-box (illustrated in the bottom graph of Figure 37) is given as the answer when the maximum entropy criterion is employed to select a particular probability distribution out of a p-box. Abbas (2003) showed that the taut string distribution is the maximum entropy distribution for the class of all distributions whose cumulative distribution functions are circumscribed by bounds such as a p-box. An example is shown in the lower, right-hand graph in Figure 38. A taut string distribution happens to also be the maximum entropy solution for the case when only the minimum and maximum of the quantity are known. This case is illustrated in the upper, left-hand graph of the figure.

But such a taut string distribution is *not* the maximum entropy solution in cases such as that depicted in the center graph in the figure where the minimum, maximum and the mean value of the quantity are known (unless the mean happens to be the average of the minimum and maximum). Instead, the maximum entropy distribution in this case is a scaled beta distribution (Lee and Wright 1994), which is generally asymmetric. The taut string distribution would have larger entropy, but it is not in the class of possible distributions to begin with because it does not have the prescribed mean. In this regard, the maximum entropy criterion can be said to be appropriately sensitive to the details of the available information about the uncertain number.

When, as in the case of Constructor, there are several disparate kinds of knowledge about an uncertain number that are drawn together, the computational problem of finding the maximum entropy distribution for any class of distributions that may emerge can be quite complex. Constructor gives only the taut string distribution. In some situations, such as most of those described in Figure 38, this distribution may not be a member of the class of distributions that has been specified by other information supplied to Constructor. (Although, when it is a member of the class, it will be the distribution with maximum entropy.) The general algorithm to select the maximum entropy distribution when disparate information is available is not known. Indeed, it is not altogether clear that the maximum entropy solution for moment and order statistics described by Jaynes

(2003) and Lee and Wright (1994) is even compatible with the maximum entropy solution for data described by Solana and Lind (1990) and Abbas (2003).

The use of the maximum entropy criterion is considered by some to be the state of the art in selecting distributions (Jaynes 2003). However, there are grave disadvantages associated with this criterion. In particular, the results of the criterion are sensitive to the scale on which the quantity is expressed. For instance, suppose we would like to apply it for an uncertain number which is a degradation rate. If we only know the minimum and maximum value of the quantity, the criterion would yield a uniform distribution. If we consider the same quantity on a different scale, we can obtain a different answer. Suppose we applied the criterion to the half life, which is clearly equivalent to the degradation rate via a simple transformation. If we only know the minimum and maximum of the degradation rate, then we clearly only know the minimum and maximum of the half life. Thus we would obtain a uniform distribution for the half life. The problem is that the uniform for the half life is not equivalent to the uniform for degradation rate by the same simple transformation. In fact, these two uniform distributions represent inconsistent and incompatible results. Such inconsistencies mean that maximum entropy cannot generally yield mathematically defensible selections.

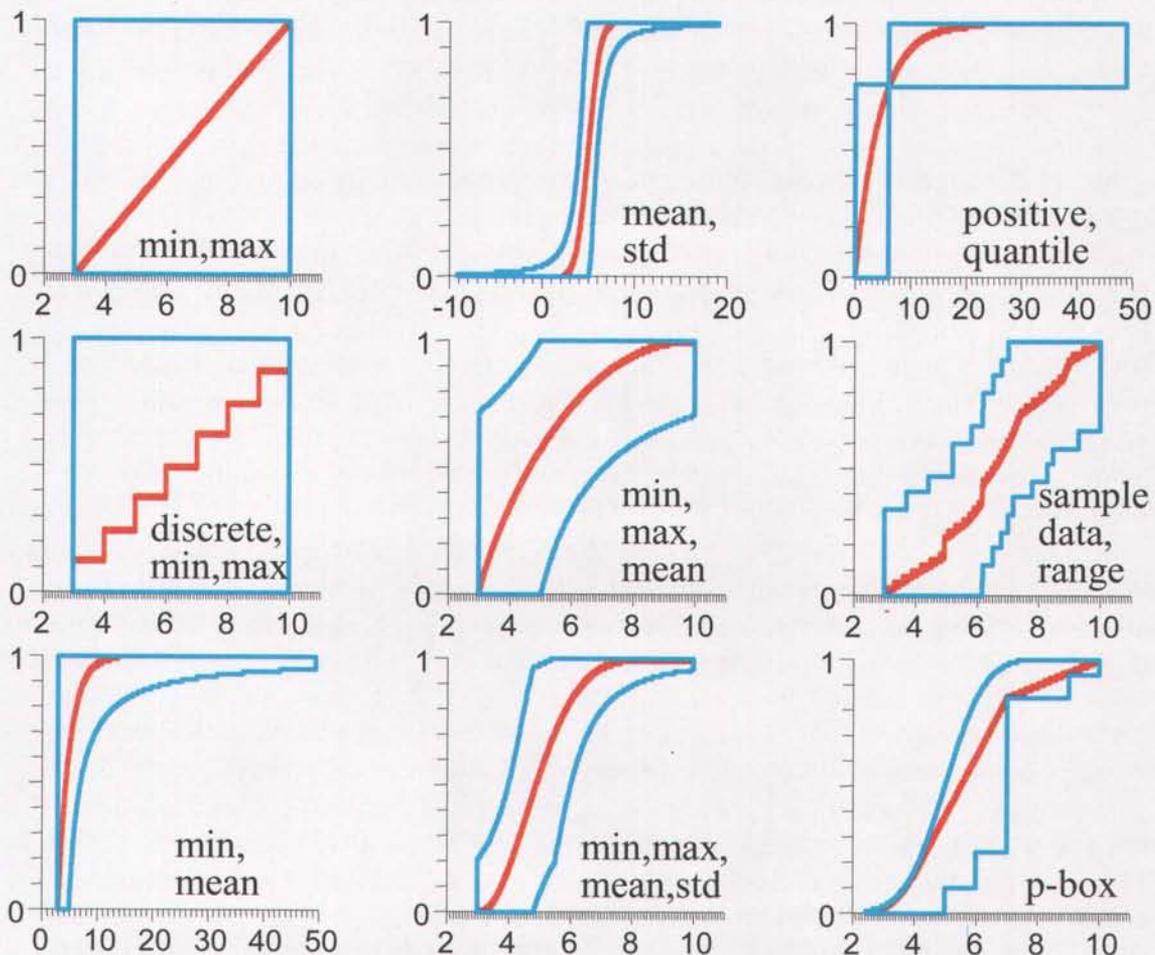


Figure 38. Maximum entropy distributions (red) for several classes of probability distribution enclosed in p-boxes (blue).

3.4 Wrangling the graphs

You can rescale, scroll and zoom in on the graphs in Constructor. The following are instructions for several common tasks.

Zooming in on the graph. You can zoom in on a portion of the graph by clicking on the upper, left-hand side of the area to which you want to zoom. While holding down the mouse button, move the mouse down and left to the lower, right corner of the area you want to highlight. A white rectangle appears to indicate the area you've chosen. Release the mouse button to zoom in on this rectangle.

Scrolling the graph. You can scroll the graph by right-clicking in the interior of the graph and, while holding down the right mouse button, move the mouse so that the graph scrolls. Right clicking is also used to invoke a popup menu. If the menu pops up and obscures the plot, just (left) click anywhere but on the menu to make it disappear.

Rescaling. You can rescale the graph by scrolling and zooming in as described above. The minimum and maximum (or truncate above and truncate below) inputs determine the range of the graph. You cannot alter the number of tic marks, or the font of the graph labels.

Restoring the graph to its original form. You can restore the graph to its form before any zooming, scrolling or rescaling, by clicking in the interior of the graph. While holding down the mouse button, move the mouse up and to the left or right so you see a white rectangle. Release the mouse button to restore the graph to its original scale.

Changing line colors and thickness. You can change the background color and the colors and thickness of the lines in the main graph (at the top of the display). To do so, select Input/Options from the main menu and modify the settings in the "Graphical display" settings. The "export" settings correspond to the temporary depiction of the pinched output distribution that is created when the Export button is clicked.

Copying the graph to the clipboard. You can put the graph on the clipboard as an Enhanced Metafile (EMF) by right-clicking it and selecting "Copy graph to clipboard". You can also copy the graph to the clipboard by double-clicking anywhere on the graph.

Printing the graph. You can print the graph by right-clicking on it and selecting Print graph. Select the printer to use and its properties via the File/Printer Setup menu option.

Moving a point. On the Density and Graph pages you can graphically edit the (lower) plot. To do this, just click on a point and, while holding down the mouse button, reposition the mouse to the spot where you want the point to be. When you release the mouse button, the graph will be redrawn with the point in its new location. When you click on a point, the cursor will change from a simple white arrow to a white arrow and a little white box.

4. Handling imprecise data

This section considers how imprecise sample data can be synthesized into an appropriate characterization as an uncertain number. The methods described here apply only to empirical sample data that would be entered in Constructor on the Data page. The discussion extends that contained in Section 3.5.4 of Ferson et al. (2003).

There are several ways that imprecise data may be synthesized into an uncertain number (i.e., an interval, probability distribution, probability box, random set, or Dempster-Shafer structure on the real line). The different methods make different assumptions about the sampling process that gave rise to the data. The synopsis below constructs p-boxes for two hypothetical data sets: one with precise numbers and one with intervals that represent measurement uncertainty.

To make the discussion concrete, we will suppose that the first data set consists of the seven values 57, 15, 76, 37, 55, 11, and 23, and that the second data set consists of the seven intervals [50, 63], [10, 18], [75, 78], [23, 48], [50, 59], [9, 11], and [20, 28]. The second data set is similar to the first except that it retains the “plus or minus” information reported by the observer which was thrown away in the first data set in favor of reporting only the point estimates. The values in both of these data sets will be understood to be random samples of some variable, where the word ‘random’ implies the values are independent and identically distributed. Most of the methods summarized here can also be applied with weighting schemes that can account for non-random sampling.

4.1 Stochastic mixture

A stochastic mixture forms a distribution from the collection of data values assuming that each can be drawn with equal probability. The resulting distribution is also known as the empirical distribution function (EDF). This distribution function is

$$S_n(x) = \frac{\#(x_i \leq x)}{n}$$

where # denotes the number of sample values that are less than or equal to a given value x and n is the sample size. In the risk analysis literature, this approach is often the preferred method for synthesizing empirical data in a distribution. Aggregating the data as a stochastic mixture is completely appropriate if the data are an exhaustive sampling of the population. The graph in Figure 39 below depicts the stochastic mixture for the first hypothetical data set of scalars. In this and all graphs in this section, the abscissa is the random variable, and the ordinate is cumulative probability.

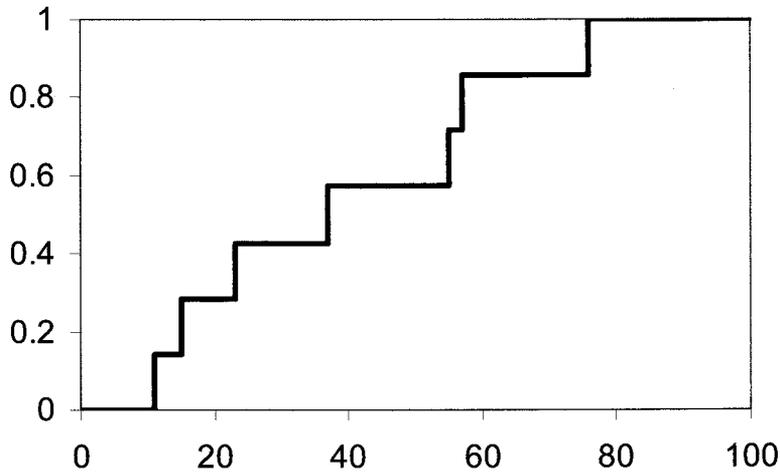


Figure 39.

The stochastic mixture yields a discrete distribution, which implies that only values already observed in the data could be sampled in future draws. The probability for each data value is the reciprocal ($1/n$) of the number of values sampled.

The graph in Figure 40 below depicts the stochastic mixture associated with the second hypothetical data set of interval values. This graph is formed as the envelope of two distributions, the first of which is the stochastic mixture of the lower bounds of each of the intervals, and the second of which is the stochastic mixture of the upper bounds of each of the intervals.

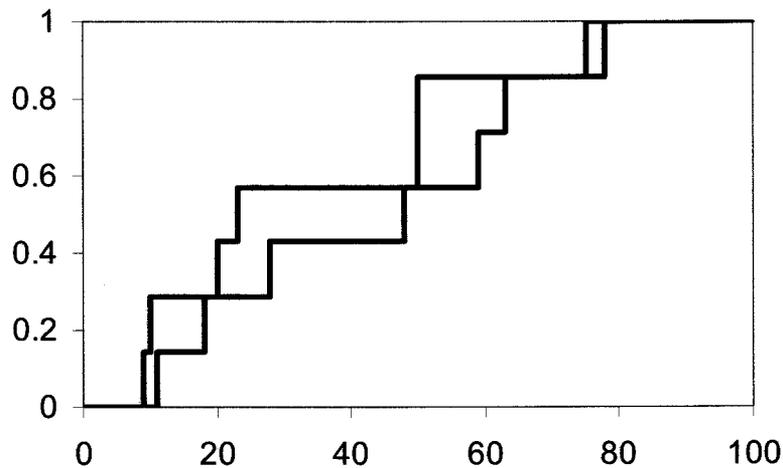


Figure 40.

Although the resulting probability box has corners, it is not really a discrete distribution because it does not assume that the probability mass is concentrated into several point values. The p-box it forms represents the class of all distributions S_n that could result from seven precise values located within their respective seven intervals.

Stochastic mixtures are obviously the tightest uncertain numbers that capture the observed variation manifested in the sample data. Although they can represent

measurement uncertainty characterized by interval data, stochastic mixtures cannot account for any uncertainty about the sampling process that produced the data.

4.2 Solana and Lind's sample rule

Solana and Lind (1990) describe a "sample rule" which holds that n independent samples of a random variable divide the real line into $n+1$ segments of equal probability. This idea is very old in probability. It is obviously very similar in spirit to the idea behind the stochastic mixture, except that we ascribe the probability masses to the intervals that partition the real line formed by the data rather than the data points themselves. The distribution produced by this method is the maximum entropy (Jaynes 2003) solution corresponding to a random sample of data points (Solana and Lind 1990; Abbas 2003).

In the case of the first hypothetical data set of scalar values, the assertion of equiprobability within each of the resulting eight segments of the real line is represented by the probability distribution shown in the graph below. Consistent with the maximum entropy criterion, this distribution assumes equiprobability *within* as well as among the segments of the real line as shown in Figure 41.

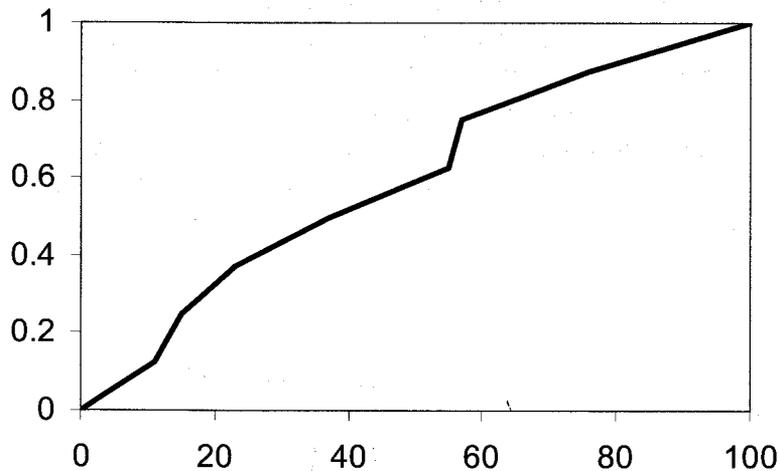


Figure 41.

This distribution is essentially an equiprobable mixture of little uniform distributions over each of the eight segments of the real line. In this case, we have assumed that the theoretical range of the random variables is known and finite. Thus the tails of the distribution are likewise little uniform distributions from the smallest possible value to the smallest observed value, and from the largest observed value to the largest possible value. Of course, the tails could in principle go out to infinity in one or both directions. If the theoretical range is unbounded on either side, Abbas (2003) suggests using a little exponential distribution from the extremal observed value to infinity.

Like stochastic mixtures, Solana-Lind maximum entropy distributions do not fully account for uncertainty about the sampling process that generated the data. When used as models of variation, the stochastic mixture presumes that all values come from the collection of values already observed. The Solana-Lind maximum entropy distribution presumes that values can come from anywhere over the range, but with probabilities perfectly prescribed by the observed data. Can this be reasonable? Random samples

from a continuous distribution divide the real line into segments that *on average* have equal probability, which is to say that, if we compute Solana-Lind distributions for a bunch of data sets each having n random samples, and ask where the $(n+1)^{\text{th}}$ sample for each data set lies, we will find it equally likely to be in any of the segments. However, it is not true that for any particular data set that we might observe we can count on the realized probabilities of future samples to be equal for the $n+1$ segments of the real line. On average, yes, but not necessarily for any particular data set.

The Solana-Lind method can be generalized to accept interval data in a natural way by finding the envelope of all possible distribution functions that could be formulated via the Solana-Lind method assuming that the data are point values within their respective intervals. This turns out to be easy to do. As was true for generalizing the stochastic mixture for intervals, we only need to compute distributions for the set of lower bounds on the data intervals and then again for the upper bounds on the data intervals. The graph in Figure 42 below depicts the p-box of the Solana-Lind maximum entropy distributions resulting from the second hypothetical data set of intervals.

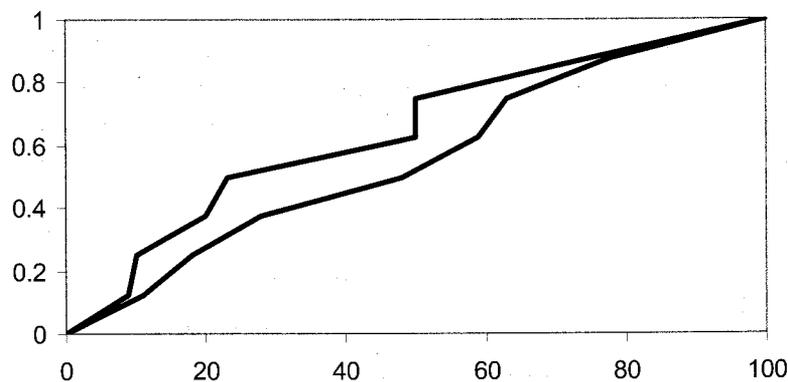


Figure 42.

Purists might suggest that this p-box should itself be replaced by a single, precise distribution representing the maximum entropy distribution from this class of distributions circumscribed by the p-box. It is possible to compute such a distribution using the “taut string” algorithm of Abbas (2003). We note that the result would not be the same as (nor even very similar to) the maximum entropy distribution obtained from the point data in the first data set.

4.3 Relaxed sample rule

A p-box that represents a weaker interpretation of the sample rule is shown in the graph in Figure 43 below. This p-box also asserts that there is equal probability in each of the eight segments of the real line (the upper and lower bounds are pinched together at these points at each step corresponding to an increase in probability by $1/8$), but it makes no claim about *where* within each of the segments the $1/8$ probability lies. The result is therefore just a Dempster-Shafer structure, and its graph is the p-box shown in the figure. Unlike the previous methods considered, this approach leads to a non-degenerate p-box even with precise sample value because it does not specify how the masses are distributed within the intervals.

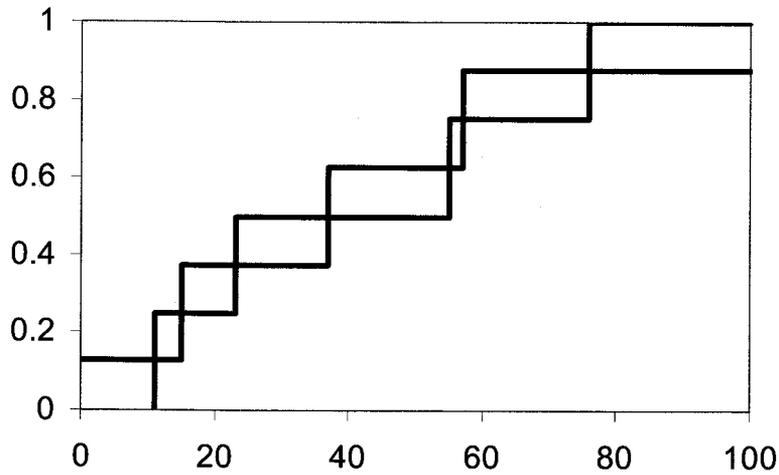


Figure 43.

This p-box is a relaxation of the precise distribution resulting from the Solana-Lind approach based on the sample rule. The relaxed sample rule can easily be generalized for use with interval data. The graph in Figure 44 below depicts its application to the second data set of interval values.

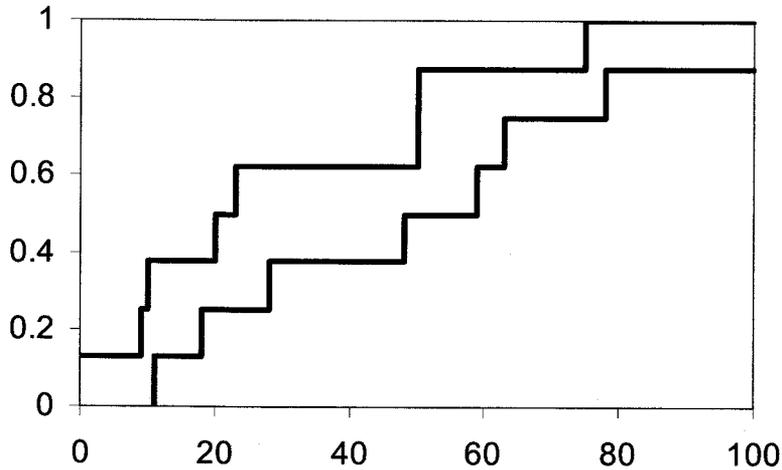


Figure 44.

Note that this p-box is somewhat wider than the analogous uncertain number developed under the sample rule and the maximum entropy criterion. Nevertheless, the relaxed sample rule does not fully account for the uncertainty about the sampling process that gave rise to the data because, like the Solana-Lind approach, the relaxed sample rule approach assumes equiprobability between the intervals, which is only true on average and not for every data set.

4.4 Kolmogorov-Smirnov

A p-box that makes a still weaker statement about the distribution for a data set is shown in Figure 45 below. This p-box represents the upper and lower 95% Kolmogorov-Smirnov confidence bounds on the distribution from which the seven data values were sampled. The breadth of these bounds acknowledges the small sample size of the data on which they are based. As the number of samples increases, the bounds tend to become closer together. The calculation of the Kolmogorov-Smirnov confidence bounds assumes that the data are random (that is, independent and identically distributed). The result is distribution-free except that it assumes continuity. Like other bounds based on confidence procedures, these bounds are not rigorous. This means there is no guarantee that the true distribution from which the values were drawn is actually entirely within the Kolmogorov-Smirnov limits. Nevertheless, they do seem considerably more reasonable than any of the previous representations of the data in view of the smallness of the sample size.

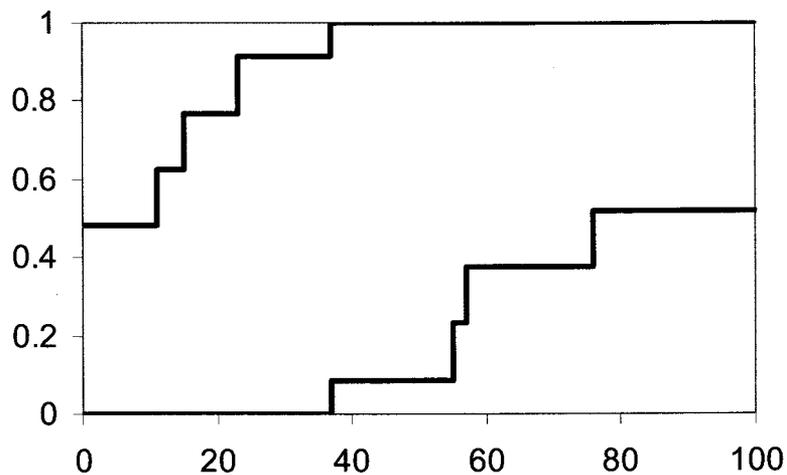


Figure 45.

The Kolmogorov-Smirnov confidence limits are computed from the stochastic mixture S_n by the formula $S_n \pm D_{\max}(\alpha, n)$, where the second term is a constant based on the sample size n and the confidence level α . Note that, for a fixed sample size, the left and right tails of the limits in the graph above do not converge to zero and one. Instead, they extend to infinity in both directions at a level that is determined by the sample size. The more samples, the smaller the tail probabilities. At the 95% confidence level, the thickness of the tails for seven samples is 0.483. If there were 100 samples, their thickness would be 0.136; for 1000, 0.043. Although this p-box's tails are theoretically infinite, they have been truncated at zero and one hundred in the graph, which would be reasonable if the quantity is known to be constrained to the range.

The Kolmogorov-Smirnov method can be easily extended for use with interval data. The graph in Figure 46 below depicts the resulting p-box computed from the second data set of interval values.

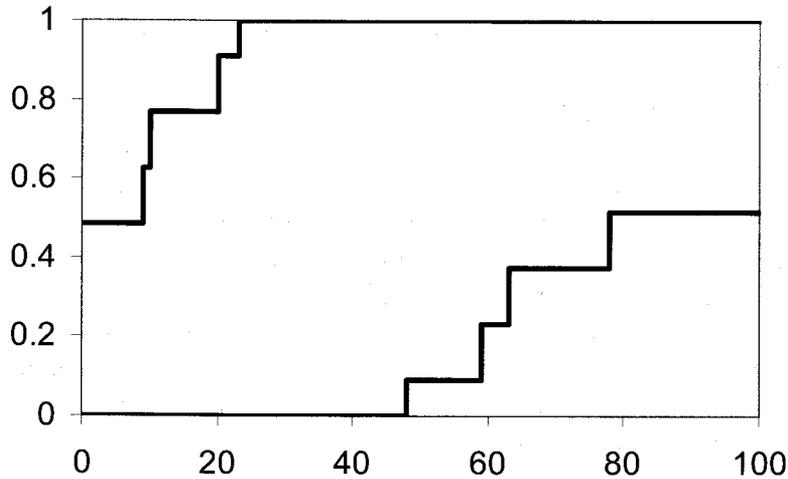


Figure 46.

This p-box is a bit wider than the previous based on point data. However, it is apparent in this case that the uncertainty arising from the smallness of the sample size contributes more to the overall uncertainty (as measured by the area between the upper and lower bounds) than does the original measurement uncertainty manifested in the widths of the data intervals.

4.5 Saw-Yang-Mo

The Chebyshev inequality gives bounds on the cumulative distribution function of a random variable for which the population mean and variance are known. Saw, Yang and Mo (1984, 1988; Ferson et al. 2003) generalized this result to handle the case where only sample estimates are available for these two moments. These bounds represent *average* limits on the distribution given the sample data (Kreinovich 2004). Although the analogy between the classical Chebyshev result and the result due to Saw et al. is therefore not perfect, it does allow us to formulate bounds on a distribution function which are much weaker than the maximum entropy formulations.

Using the Saw-Yang-Mo method on the first data set of scalar values produces the p-box depicted below. This p-box depends only on the sample mean and variance and no other characteristics of the sample data, so it does not make use of all the available information from the sample. In this case, the sample mean and standard deviation are 39.14 and 24.39 respectively. These values yields the bounds depicted in the graph in Figure 47 below.

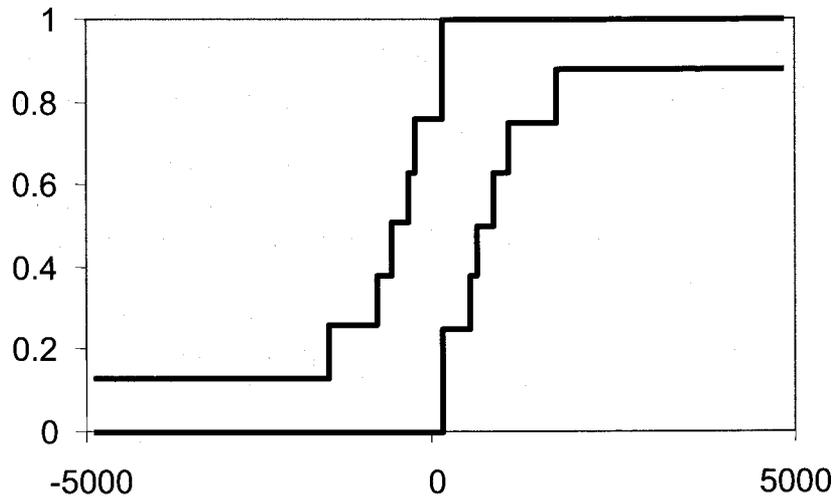


Figure 47.

Like the Kolmogorov-Smirnov limits, these are not rigorous bounds, thus, despite their great breadth, there is no guarantee that they will enclose the true distribution, as can be shown by the construction of counterexamples (Kreinovich 2004).

Note that both the left and right tails extend to infinity in either direction. They are truncated in the graph at ± 5000 for the sake of the display. The thickness of the tails is related to the number of samples. With only seven samples, percentiles smaller than the 12th and larger than the 88th are unbounded. If there were 19 samples, the tails would be thinner and all percentiles between the 5th and 95th would be bounded. If there were 100 samples, we would have bounds up, but not including, to the 1st and 99th percentiles.

We can also apply the Saw-Yang-Mo method to the second data set of intervals. In this case the sample mean is the interval $[33.86, 43.57]$ and the sample standard deviation is the interval $[21.92, 28.40]$. The resulting Saw-Yang-Mo p-box is depicted in Figure 48.

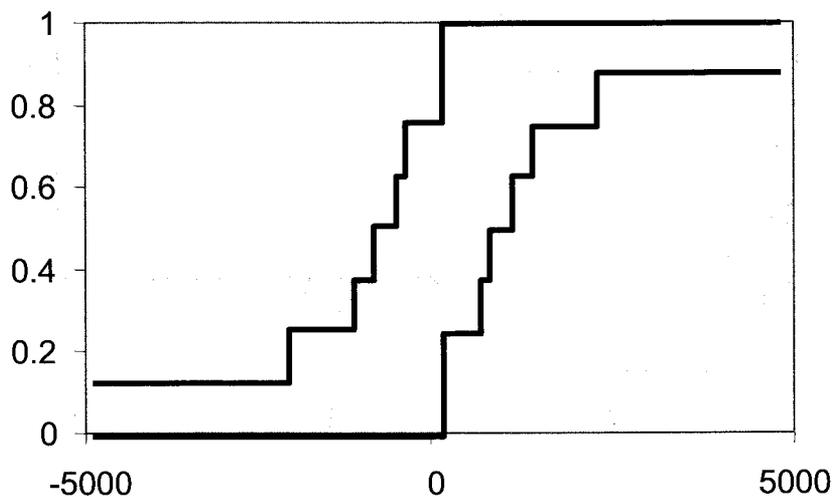


Figure 48.

Note that this p-box is wider than the p-box based on data given as point values, although the contribution to overall uncertainty from the widths of the data intervals is very small compared to the sampling uncertainty arising from having only seven samples. Although the p-boxes shown here are quite wide, it is interesting to note that they do not strictly enclose the corresponding Kolmogorov-Smirnov p-boxes.

If the value of the variable is limited to some specific range, such as when the variable is a percentage and the range is the interval between zero and 100, then the bounds can be constrained to the range. For instance, if we assume that the two data sets are estimates of percentages, then the resulting p-boxes resulting from the Saw-Yang-Mo method are those depicted in Figure 49 below. The upper graph is based on the first data set of point values, and the lower graph is for the interval data.

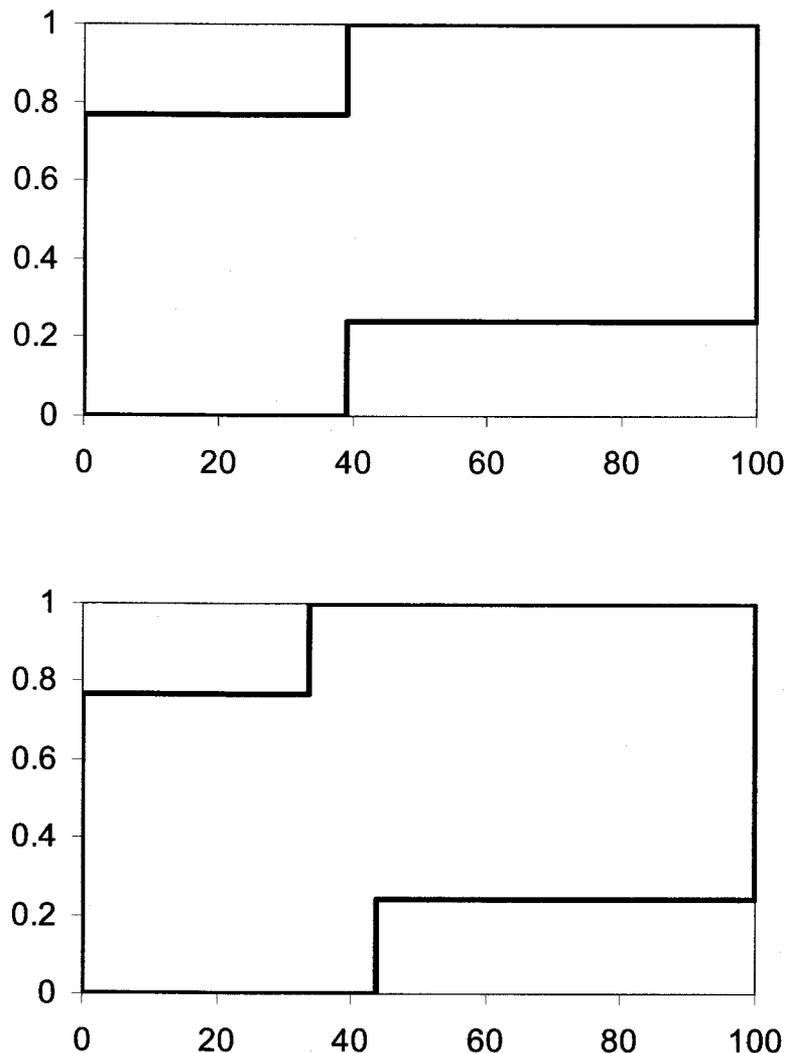


Figure 49.

These p-boxes are simply truncations of the previously depicted Saw-Yang-Mo p-boxes. It would be useful to have a generalization of the Saw-Yang-Mo inequality that

accounts for the case when the range of the variable in question is finite and known. Research in this direction might be productive. The resulting p-boxes would be expected to be tighter than those depicted here.

A salient disadvantage of the Saw-Yang-Mo approach is that the bounds it produces do not converge to the true distribution when sample size and precision are large. Instead, they converge to the Chebyshev limits given by parametric information about the population's mean and dispersion.

4.6 Rigorous bounds from sample data

If there are only seven members of the population, then the seven values in the data set would represent an exhaustive sample of the population. If this is the case, then S_n is *the* distribution for this population. There would be no need for rigorous bounds as the distribution would be exact. If, on the other hand, there are ten members of the population, then the data set would be only a subset of the ten values. What could one conclude about the population's distribution in this case? What are the possible distribution functions for the population?

There are two extreme situations. The first one arises when we assume all the unsampled points are smaller than the smallest observed value. The second extreme situation arises when we assume the unsampled values are all larger than the largest value observed. Suppose there are $m+n$ members of the population, of which n have been sampled. Because the unsampled values could be arbitrarily small or large, the bounds on the distributions for the extreme situations would be

$$B_{n,m}(x) = \left[\frac{\#(x_i \leq x)}{m+n}, \frac{m + \#(x_i \leq x)}{m+n} \right]$$

where, again, # denotes the number of sample values smaller than or equal to a given value x . For the seven points of the first data set of point values, these bounds are depicted in Figure 50 below.

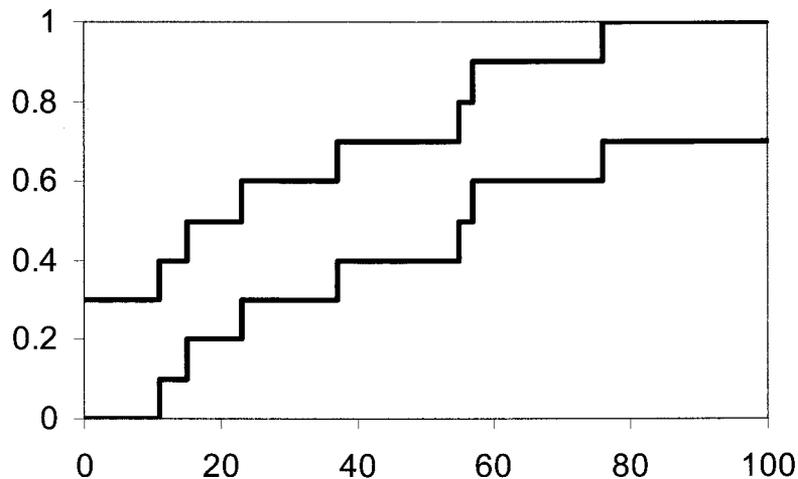


Figure 50.

The bounds would be slightly wider for the second data set of interval values. The uncertainty about the tails is $m/(m+n) = 3/10$. This uncertainty extends to infinity, or the possible range of the variable, in both directions. The depicted bounds are rigorous bounds given the seven data values and the fact that the entire population has only ten members. However the three other values might be sprinkled along the real line, the distribution for the entire population is sure to lie within these bounds.

The thickness of the tails is obviously a function of how many members of the population have not been sampled. Consequently, as the sample gets smaller or the population is infinite, this approach allows us to say less or nothing about the distribution function. There is no nontrivial method that can infer *rigorous* bounds on a distribution of an infinite population without making use of fairly strong assumptions. In practice, even moderate assumptions yield bounds that can be very wide for small sample data. Of the methods considered here, the Kolmogorov-Smirnov method seems to yield the most reasonable characterization of the uncertainty about a distribution function based on finite sampling from an infinite population. Unlike stochastic mixture, the Solana-Lind method, and the approach based on the relaxed sample rule, Kolmogorov-Smirnov method accounts for the uncertainty about the sample process in a way that yields tighter bounds for cases with more samples. And, unlike the Saw-Yang-Mo method, when sample size becomes very large, the bounds converge to the true distribution. Other than assuming continuity of the distribution, the Kolmogorov-Smirnov confidence limits are a distribution-free result, meaning that they make no assumption about the shape of the underlying distribution. The approach assumes only that the data samples are random. The Kolmogorov-Smirnov method can be generalized to account for interval measurement uncertainty in straightforward way. Amendments to the method to account for non-randomness in the sampling design are currently under investigation. Also worthy of further research are ways to account for the finiteness of the variable's range that could improve the bounds in the tails.

In empirical assessments, there can be a tradeoff between the number of samples that can be collected and the precision of individual samples. Although some statisticians seem to suggest that it is always preferable to increase sample size rather than improve precision, this is clearly not always true. Any method that hopes to recommend the proper sampling strategy to minimize overall uncertainty obviously must discern between variability and incertitude and yet must also be able to synthesize them both into a general measure of uncertainty. The p-box produced by the Kolmogorov-Smirnov method is a clear candidate for such a method.

The Kolmogorov-Smirnov method shares the limitations of other methods based on confidence procedures. The main limitation is that they do not allow us to make a definitive statement about the underlying variable. We don't even know the probability that variable is within the interval. Suppose that we compute, say, a 95% confidence interval about some statistic. And let's suppose that all of the requisite assumptions for this calculation are actually true, that the samples really were random, etc. Scientists would sometimes like to think that we can conclude that there is 95% probability that the next random value is within the confidence interval. Statisticians know, however, that we cannot make this inference. It is simply not true, and it is easy to construct simple counterexamples to show it is not so. The fundamental reason is that the true value of the

statistic in any particular situation is a constant value, rather than a random variable. It is either inside the interval or it's not. According to the frequentist* interpretation, it doesn't make sense to speak of the probability of such an event.

Because confidence limits do not represent sure statements, they cannot be used in interval or other bounding analyses without specific assumptions. Likewise, because there is no guarantee that the true population distribution will lie entirely within the Kolmogorov-Smirnov limits, if subsequent calculations done presume this rigor, then one must make an additional assumption that it is so. In other words, because the Kolmogorov-Smirnov limits are not rigorous, analysts may need to make a specific assumption that the limits enclose the true distribution.

On the other hand, the distribution-free nature of the Kolmogorov-Smirnov method may result in limits that are actually too conservative. This method only makes use of sample data, and is insensitive to any other kind of information that an analyst may have about an uncertain number. The design of Constructor allows the user to bring a variety of kinds of information together and compose from the assemblage a coherent characterization of the uncertain number. For instance, if one knows the distribution is lognormal in form, then entering this fact on the Shape page may help to tighten the bounds. Likewise, if one knows constraints on probability densities, then specifying these constraints on the Density page may further tighten the bounds on the uncertain number. In general, the p-box characterizing sample data can be combined with constraints on all the other pages of Constructor.

*Subjectivist methods purport to allow us to compute such a probability. This is what the Saw-Yang-Mo method does, for example. It gives us bounds on the probability that the next value to be sampled will be larger than some value. Unfortunately, the answer it produces is not guaranteed to be *correct* in the sense that if we iterated the situation many times we would discover empirically that the value actually was in the interval $100(1 - \alpha)\%$ of the times. Subjectivist methods can compute these probabilities because they interpret *probability* to be something in one's head, rather than an empirical fact about the world. Deborah Mayo (1996) likens making these subjectivist calculations to balancing a checkbook without ever confirming the results against a bank statement. I can *believe anything* about my balance, but believing something does not make it so. Statisticians have given us two approaches with which to make inferences from sample data. The first (frequentist) approach, when followed strictly, can be shown to yield correct results in some cases, but in many situations it doesn't give any answer at all. The second (subjectivist) approach will always give an answer, but the answer it gives may not be correct.

5. Options and program settings

Selecting Input/Options from the main menu brings up an options dialog like Figure 51.

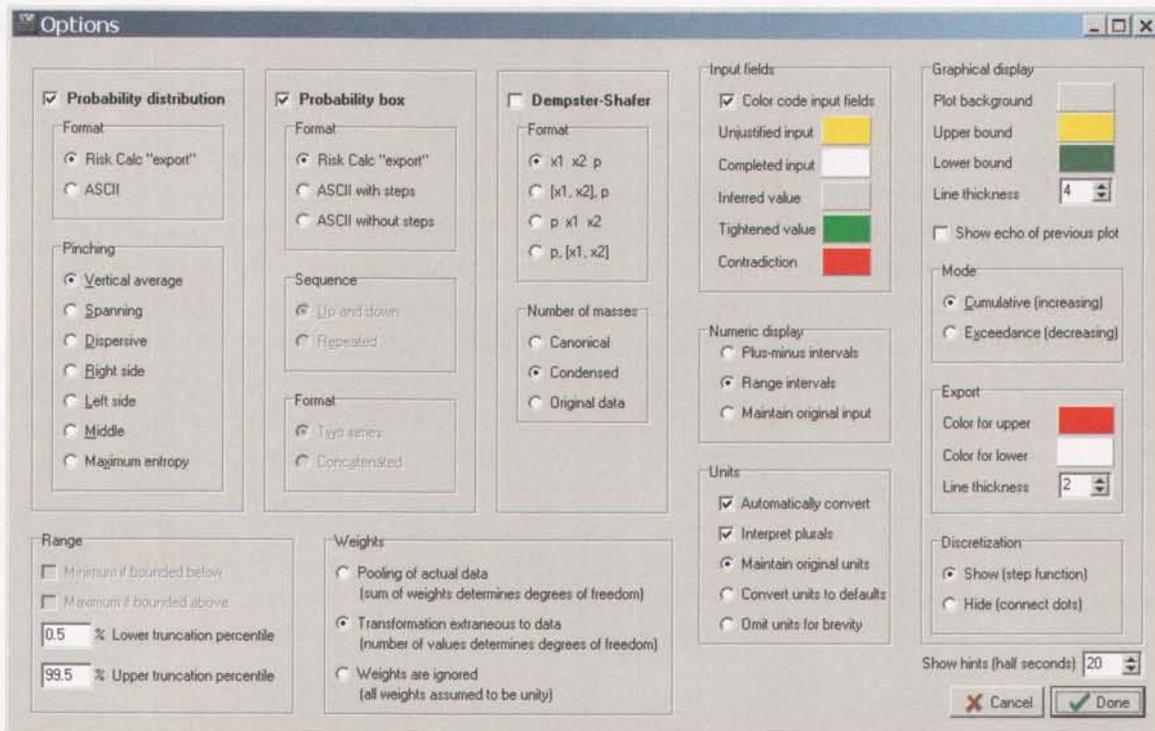


Figure 51.

The most important choice you make from this dialog is which kind of uncertain number you want to create when the Export button on the main dialog is clicked. You can select any or all of the three checkboxes at the tops of the three vertical panels on the left of the options dialog to create a probability distribution, probability box, and Dempster-Shafer structure as output. Check the box at the top of a panel to create that output. A single click of Export produces a file for each checked output.

As has been described in the tutorials, you can customize many program features and behaviors by making other changes on this dialog. Any changes you make take effect immediately, and don't require that you click the Done button. Here is a synopsis of the available options.

Format. Various format options are available for each of the three output types. For probability distributions and p-boxes, you may elect to format output so that it can be conveniently read by RAMAS Risk Calc (Ferson 2002). Alternatively, a simple ASCII text output may be used. You can format a p-box as a step-function or connect-the-point bounds. (If you want to see the difference between the step-function and the connect-the-point bounds, use the Discretization option in the lower, right corner of the options dialog). For Dempster-Shafer structures, there is no Risk Calc format, but you can select between outputting the x -coordinates or the masses (p 's) first, and whether to separate numbers on a line with blanks or with commas and parentheses. For p-boxes, there is a

secondary format choice. If you select “Two series”, a blank line is written between the left and right sides of the p-box. If you select “Concatenated”, no blank line is written.

Pinching. This option controls how residual uncertainty about a probability distribution is resolved. The Pinching option was described on page 59.

Sequence. These options control whether the coordinates of the p-box are output up (from 0 to 1 probability) for the left side and down (from 1 to 0) for the right side, or repeatedly (from 0 to 1 for both the left and right sides of the p-box). The former is useful for depictions of cumulative distribution functions and the latter is useful for depictions of complementary cumulative distributions or exceedance risk functions.

Number of masses. For Dempster-Shafer structures, you can select how many lines are written to the output file. Selecting “Canonical”, always writes 100 interval-mass pairs. Selecting “Condensed”, decomposes the uncertain number into horizontal slices and then combines intervals if they are identical (summing their masses). Selecting “Original data” makes Constructor regurgitate into the output file the original interval-mass pairs specified on the Data input page.

Range. Here you can specify lower and upper points at which infinite distributions will be automatically truncated. The lower must be larger than zero, and the upper must be smaller than one. These limits are specified as percentiles. If you want to specify precise x -values for truncation limits, specify them as the minimum and maximum on the Parameters input page.

Weights. Here you can choose between two interpretations for the weights you specify on the Data input page. You can choose to have the weights interpreted as sample sizes, as importance measures, or totally ignored. These options are discussed on page 39.

Input fields. With these controls, you can set the color coding used to indicate whether inputs have been justified, were inferred or tightened from other inputs, or are in contradiction with other inputs. Edit a color by clicking on it to evoke a color editor. You can change one of these colors by clicking on it to invoke a color editor. You have to close the color editor, by clicking its OK or Cancel button, before changing other options. Turn off color coding altogether by unchecking the box.

Numeric display. With these controls, you can set the format of any intervals displayed. Clicking “Maintain original input” turns off input echoing altogether and the fields remain as you enter them.

Units. You can turn on and off units conversion checking, whether a final “s” should be recognized as a plural (so meter and meters are the same), and whether and to what units should be converted for display.

Graphical display. Options in this panel control how the graph in the upper part of the main display of Constructor. (They do not change the appearance of the graphs in the

Graph or Density input pages.) The colors used to display the plot background and the upper and lower bounds of the graph are shown on little rectangles. You can change one of these colors by clicking on the rectangle to invoke a color editor. You have to close the color editor, by clicking either its OK or Cancel button, before you can change other options. You can also select the width of the lines used to draw the curves. You can also choose to have Constructor display an echo of the previously displayed graph whenever the graph is redisplayed. This can be especially useful to explore the effect of a single input you make. Your option changes become effective as soon as the graph is next redrawn. You generally do not have to close the options dialog to redraw the graph.

Mode. Selecting “Cumulative” makes the curves on the main Constructor graph monotonically increasing. Selecting “Exceedance” inverts the display so the curves are monotonically decreasing. Constructor cannot display density graphs.

Export. These options allow you to choose the color and line thickness used to display the graph of the uncertain number actually written to an output file when the Export button (on Constructor’s main display) is clicked. This “export” display appears briefly on the main Constructor graph atop any curves already on it. In the case of p-boxes, the additional display looks like highlighting. For a probability distribution, the exported uncertain number often looks rather different according to the pinching that is used as described on page 59. When Dempster-Shafer structures are exported, there is no “export” display.

Discretization. The graph in the upper part of the main display can be shown either as two step functions, or as smoother connected-the-point bounds. Because there are 100 discretization levels, the differences between these two forms are often rather subtle.

Finally, there is a lone option, just above the Done button of the options dialog, that controls how long balloon hints are displayed as you let the mouse cursor linger over an input field. Increase this time if you want to be able to examine the histories that are summarized in the hints, or decrease it if you don’t want the hints to obscure the input dialog.

6. Automatic input cascading

Cascading inputs. When you enter inputs into Constructor, the software propagates the information to partially constrain other inputs. The entries made in one field thus automatically cascade to other fields on the page, and to fields on other pages. You can turn off this feature by clicking on the Relax button or by selecting Bounding/Relaxed from the main menu. Selecting Bounding/Exploratory (see page 41) also turns it off. You can restore automatic input cascading by clicking the Constrain button, or by selecting Bounding/Constrained from the main menu.

Aggregating p-boxes among pages. By default, the information you supply on any of the tabbed pages for inputs ('Shape', 'Parameters', etc.) is combined to find the tightest justifiable bounds on the uncertain number. It may sometimes be useful to ask the program to construct a p-box using only the information given on a single page of inputs. To do this, select Bounding/Current page from the main menu. When you do, the p-boxes for the various pages are insulated from each other. This feature can be useful if the bounds cross when the information from the pages is combined. You can restore the normal behavior of the program by selecting Bounding/All pages from the main menu. You can alter the program's behavior in another way too. Instead of aggregating the bounds via intersection, you may elect to form the aggregation as an envelope or convex hull of the p-boxes from the seven input pages. To do this, select Bounding/Envelope from the main menu. This option is only available if the menu option Bounding/All pages has been selected. Aggregating via enveloping might be appropriate if the information in the different pages came from different sources and you don't know which source was correct. You can restore the normal behavior of the program by selecting Bounding/Intersection from the main menu.

Distribution shape cascading to qualitative properties. When a family of distribution shapes is selected on the Shape page, the choice cascades to inform the qualitative features listed on the left of the page. The policy used by Constructor in making the inferences is the following: If the selected distribution always has a property, then the word Yes appears next to the property. If the distribution never has the property, then the word No appears instead. If it can have the property under some parameter values, then a blank appears, unless it has the property *only* when the distribution is a scalar, in which case the word No appears. The reason for neglecting the degenerate cases of scalars is that, without this exclusion, essentially no inferences could be drawn about properties at all. Scalars are both continuous and discrete. They are simultaneously convex and concave, and they have a hazard rate that is both increasing and decreasing. In general, however, Constructor respects extreme and limiting cases. For instance, it considers a uniform to be one of the unimodal distributions although some might not categorize it as such because it is the limiting case of sequences of distributions that are.

Qualitative properties cascading to other input fields. When a qualitative feature is ascribed to the distribution of the uncertain number on the Shape page by selecting Yes or No for any of the eleven properties listed there, then the p-box at the top of the display may be tightened to exclude any distributions that would have been possible but are now

ruled out by having or not having the property. Of course, in many cases, the choice may yield no tightening of the current displayed p-box, or the tightening may be delayed until you specify other details about the uncertain number or its distribution.

Setting a qualitative property to Yes or No cascades to inform the other qualitative properties mentioned on the page. For example, setting the integer-valued property to Yes cascades to set the discreteness property to Yes and the continuous property to No. The input fields for the latter two properties are color-coded gray to reflect the fact that they were inferences, and only the integer-valued property is color-coded yellow because it was the direct input that needs justification. Setting integral-valued property to No, on the other hand, would not imply anything about either the discreteness or the continuity property. Likewise, setting discreteness to No wouldn't say anything about continuity, although it does imply the distribution cannot be integer-valued.

Setting a qualitative property to Yes or No also cascades to inform the list of possible shape families. Shapes that are disqualified by your assertion will then appear in the list shifted to the right and prefaced by the word "NOT". In this way, characterizing the qualitative features of the distribution can narrow down the selection of the distribution shape. Choosing one of the disqualified distributions would obviously be inconsistent with the property you've ascribed to the uncertain number and may precipitate a contradiction error condition.

In many cases, choosing a qualitative property will also cascade to inform parameters whose input fields appear on the Parameters or Shape pages. For some properties, the effect on parameters is obvious. For instance, saying that the distribution is integer-valued implies that the mode, median, minimum, maximum, interquartile range, and range (although not the mean, variance or standard deviation) are also integer-valued. If these parameters were integers, or intervals with integer bounds, you might not notice that the cascade occurred. For some properties, the effects on parameters may be somewhat surprising. For instance, saying the distribution is symmetric implies that the parameters are related to each other through this symmetry. For example, if you had specified the minimum and mean, setting the symmetry property to Yes enables Constructor to compute the median and the maximum. The median will of course just be the same as the mean, and the maximum will be twice the mean minus the minimum. If you specified interval bounds for these parameters, setting the symmetry property to Yes will sometimes enable Constructor to tighten some or even all of the intervals.

7. Frequently asked questions

Q. What if I don't know the value of a parameter or other input?

A. Leave it blank to indicate total ignorance. Alternatively, you can enter a (possibly wide) interval to represent what little you are sure about. To indicate an upper or lower bound only, use the $>$ or $<$ characters. For instance, entering " >0 " means that you know the number is positive.

Q. How do I enter a Dempster Shafer structure?

A. Use the Data page, entering the focal elements as intervals and the masses as weights that add up to unity. Select "Stochastic mixture" to form the Dempster-Shafer structure. If the focal elements are not closed intervals, then Constructor can only handle your Dempster-Shafer if you transform them into closed intervals, perhaps by taking their convex hulls. If the frame of discernment Ω is not the real line, then you would need to map Ω into the real numbers.

Q. Can I use Constructor to summarize sampling data from calculations?

A. Sure. Just enter the sample values on the Data page. If you have scalar values, be sure that the lower and upper values of the interval are the same.

Q. How do get a picture of the result?

A. If you double-click on the graph on the upper part of the display (or right-click on it and select "Copy graph to clipboard"), a copy of it will be placed on the clipboard in metagraphics format. You can then paste the graph into other Windows applications. You may want to modify the background and line colors via the Input/Options menu choice.

Q. I only have a very small number of sample values? What can I say about the distribution from which they are drawn?

A. Unless the underlying population is also very small, the Kolmogorov-Smirnov confidence limits (available on the Data page) are probably the best representation of your uncertainty.

Q. I entered sample values on the Data page but the Show button doesn't produce a graph? What gives?

A. You probably forgot to enter the weights. If you don't want to use weights, uncheck the "Specify weights" box.

Q. Why did the input field turn yellow after I typed in it?

A. Inputs are color coded to indicate their status. Yellow means that you've specified the numerical value but you haven't yet given a verbal justification to support the entry. To do so, right click on the input field and type in the box labeled "Justification" on the

dialog that pops up. You can also specify other information on this dialog. You can change the color used by selecting Input/Options from the main menu and clicking on the little yellow panel labeled “Unjustified input”. You can also turn off the color coding altogether by unchecking the box above the yellow panel, but this is not recommended.

Q. Why did an input field turn red?

A. The red color coding is telling you that this input contradicts another input you made. You may want to click on the Reset button (in the lower, left-hand corner of the display) and then click on the Yes button to clear all the entries you’ve made on the page. You can change the color used to indicate contradictions by selecting Input/Options from the main menu and clicking on the red panel labeled “Contradiction”.

Q. My inputs have over-determined the uncertain number and created contradictions. What should I do?

A. Evidently you don’t know as much as you thought you did about the uncertain number. You need to relax one or more of the inputs. You may want to click on the Reset button (in the lower, left-hand corner of the display) to clear entries you’ve made. If you want to change some of your inputs but the program keeps resetting them, it may help to press the Relax button, or select Bounding/Relaxed from the main menu. This turns off the automatic cascading that Constructor uses to propagate your inputs. It can also be useful to select Bounding/Current page to have Constructor build bounds on the uncertain number based only on the information you’ve specified on the current tabbed page of inputs. Alternatively, you may select Bounding/Envelope if you still want to have Constructor use information from all the pages.

Q. How do I keep from getting tangled up in a knot of red color coding and error messages about bounds crossing or left bounds being greater than right bounds?

A. The red color coding and these error messages are telling you that your inputs contradict each other. If you’re just browsing and want to see what the program can do, select Bounding/Exploratory from the main menu. This will turn off the color coding and the automatic cascading of inputs so that you can change inputs as you like without having to make them all agree with each other. Buttons labeled “Test” will appear in the lower, right-hand corners of several of the pages. Clicking on the Test button will put some example inputs in the current screen for you.

Q. How do I turn off Exploratory mode?

A. You can’t. You need to close the program (by selecting File/Exit from the main menu) and invoke Constructor again afresh. The reason for this is that, when you are exploring, the program is still accumulating histories (those hints that appear when the mouse lingers over an input field) and keeping track of information that you don’t want to be part of any actual session record. Restarting the program from scratch gets rid of this extraneous information. Although you can’t turn off the exploratory mode, you can turn cascading back on (by selecting Bounding/Constrained from the main menu) and restore color coding (by checking the box labeled “Color code input fields” on the options dialog which you invoke by selecting Input/Options from the main menu).

Q. How can I see bounds on the probability *density* distribution?

A. The bounds on the probability density are displayed in the lower graph (not Constructor's main graph) on the Density page, which you can access by selecting Input/Density from the main menu or just by clicking on the Density tab. Double-clicking on the graph displays data entry spreadsheets. Double-clicking on either spreadsheet restores the lower density graph. You can only see bounds on the probability density distribution if you enter some on the Density page. Only the bounds you specify, either interactively with the mouse on the lower graph or by entering numbers on the spreadsheets affects the bounds on the density distribution.

Q. Why doesn't the information I specified on other pages propagate to the density graph?

A. The information you may have entered on the other pages, generally says little or nothing about the probability density. There are, in fact, no non-trivial bounds on density implied by bounds on the cumulative distribution (unless those bounds are coincident). It's easy to see why. As long as the upper and lower bounds on the cumulative are different for a value x , they imply there could be a finite amount of mass located at the point x . If this is true then the density at the point x is infinite. At the same time, if the upper and lower bounds for a percentile include x and are not different, then this implies there might be zero mass at x . Thus, upper and lower bounds on the density would be between zero and infinity, which are trivial bounds.

Q. If " >0 " means greater than zero, how do I express greater than *or equal to* zero?

A. The expression " >0 " serves for both. For practical purposes, it is unnecessary to make this distinction for probability distributions and other uncertain numbers that are in computer representations. The expression allows all values that are larger than zero by any amount, including amounts that are too small to represent on a computer. Technically, therefore, the expression merely means that the value cannot be any smaller than zero. The alternate syntax would have been $[0, \infty]$, but this is longer and it's not easy to type the symbol ∞ in Windows. All the intervals you specify in Constructor are assumed to be closed (i.e., include their endpoints).

Q. Why does vertical averaging over ranges of x where the upper and lower bounds are horizontal result in a pinched distribution that seems to be slightly increasing (as in graph B of Figure 36)?

A. Although the upper and lower bound are expressed with two sequences of points and outward-directed rounding, the exported distribution is specified with only a single sequence of points. The red curve connects the dots between these points and this induces the slight upward slant.

8. References

- Abbas, A.E. (2003). Entropy methods for univariate distributions in decision analysis. *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 22nd International Workshop*, edited by C.J. Williams, American Institute of Physics.
- Ayyub, B. M., (2002). *Elicitation of Expert Opinions for Uncertainty and Risks*, CRC Press, Boca Raton, Florida.
- Ayyub, B.M. and R. McCuen (1997). *Probability, Statistics and Reliability for Engineers*. CRC Press, Boca Raton, Florida.
- Berleant, D. and J. Zhang (2004). Representation and problem solving with Distribution Envelope Determination (DEnv). *Reliability and Engineering System Safety* 85: 153-168.
- Bernat, G., A. Burns, and M. Newby (2004). Probabilistic timing analysis: an approach using copulas. *Journal of Embedded Computing* [to appear].
- Caselton, W. F. and W. Luo (1992). Decision making with imprecise probabilities: Dempster-Shafer theory and application. *Water Resources Research* 28(12): 3071–3083.
- Dempster, A.P. (1967). Upper and lower probabilities induced by a multi-valued mapping. *Annals of Mathematical Statistics* 38: 325–339.
- EPA [U.S. Environmental Protection Agency] (2002). *Calcasieu Estuary Remedial Investigation/Feasibility Study (RI/FS): Baseline Ecological Risk Assessment (BERA)*. Available on-line at <http://www.epa.gov/eart1r6/6sf/sfsites/calcsinit.htm>; <http://www.epa.gov/earth1r6/6sf/pdf/appendix1.pdf>, and <http://www.epa.gov/earth1r6/6sf/pdf/appendix2.pdf>.
- EPA [U.S. Environmental Protection Agency] (2003a). *Ecological Risk Assessment for General Electric (GE)/Housatonic River Site, Rest of River*. DCN GE-070703-ABRC.
- EPA [U.S. Environmental Protection Agency] (2003b). *Human Health Risk Assessment for General Electric (GE)/Housatonic River Site, Rest of River*. DCN GE-031203-ABMP.
- Ferson, S. 2002. *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*. Lewis Publishers, Boca Raton, Florida.
- Ferson, S. and J.Hajagos (2004). Arithmetic with uncertain numbers: rigorous and (often) best possible answers. *Reliability and Engineering System Safety* 85: 135-152.
- Ferson, S. and W.T. Tucker (2003). Reliability of Risk Analyses for PAH-contaminated Groundwater. In *Groundwater Quality Modeling and Management Under Uncertainty*, S. Mishra (Ed.). Proceedings of the Probabilistic Approaches & Groundwater Modeling Symposium held during the World Water and Environmental Resources Congress in Philadelphia, Pennsylvania, June 24-26, 2003. Reston, VA: ASCE Publications.
- Ferson, S., V. Kreinovich, L. Ginzburg, K. Sentz and D.S. Myers (2003). *Constructing Probability Boxes and Dempster-Shafer Structures*. Sandia National Laboratories, Technical Report SAND2002-4015, Albuquerque, New Mexico. Available at <http://www.sandia.gov/epistemic/Reports/SAND2002-4015.pdf>.
- Fetz, T. and M. Oberguggenberger (2004). Propagation of uncertainty through multivariate functions in the framework of sets of probability measures. *Reliability Engineering and System Safety* 85: 73-88.

- Gigerenzer, G. (2002). *Calculated Risks: How to Know When Numbers Deceive You*. Simon & Schuster, New York.
- Goovaerts, M.J., J. Dhaene, and A. De Schepper (2000). Stochastic upper bounds for present value functions. *Journal of Risk and Insurance* 67: 1-14.
- Halpern, J.Y. (2003). *Reasoning about Uncertainty*. MIT Press, Cambridge, Massachusetts.
- Hammond, K.R. (1996) *Human Judgment and Social Policy: Irreducible Uncertainty, Inevitable Error, Unavoidable Injustice*. Oxford University Press, New York.
- Helton, J.C., J.D. Johnson and W.L. Oberkampf (2004). An exploration of alternative approaches to the representation of uncertainty in model predictions. *Reliability and Engineering System Safety* 85: 39-72.
- Henrion, M. and B. Fischhoff (1986). Assessing uncertainty in physical constants. *American Journal of Physics* 54(9): 791-798.
- Isbell, D., M. Hardin and J. Underwood. (1999) Mars Climate Orbiter team finds likely cause of loss. <http://mars.jpl.nasa.gov/msp98/news/mco990930.html>.
- Jaynes, E.T. (edited by G. Larry Bretthorst) (2003). *Probability Theory: The Logic of Science*. Cambridge University Press.
- Johnson, P.E., F. Hassebrock, A.S. Duran and J.H. Moller (1982). Multimethod study of clinical judgment. *Organizational Behavior and Human Performance* 30: 201-230.
- Kaas, R., J. Dhaene, and M.J. Goovaerts (2000). Upper and lower bounds for sums of random variables. *Insurance: Mathematics and Economics* 27: 151-168.
- Klir, G. and B. Yuan (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, New Jersey.
- Kozine, I.O. and L.V. Utkin (2004). An approach to combining unreliable pieces of evidence and their propagation in a system response analysis. *Reliability Engineering and System Safety* 85: 103-112.
- Kriegler, E. and H. Held (2005). Utilizing belief functions for the estimation of future climate change. *International Journal of Approximate Reasoning* 39: 185-209. Available at http://www.pik-potsdam.de/~kriegler/docs/kh_belief_climate.pdf.
- Kreinovich, V. et al. (2004). *Approximating Uncertainty about Output from Black-box Functions Whose Inputs Are Characterized by Uncertain Numbers*. Applied Biomathematics Technical Report, Setauket, New York.
- Lee, R.C. and W.E. Wright (1994). Development of human exposure-factor distributions using maximum-entropy inference. *Journal of Exposure Analysis and Environmental Epidemiology* 4:329-341.
- Luo, W. B. and B. Caselton (1997). Using Dempster-Shafer theory to represent climate change uncertainties. *Journal of Environmental Management* 49: 73-93.
- Matheron, G. (1975). *Random Sets and Integral Geometry*. Wiley, New York.
- Mayo, D. 1996. *Error and the Growth of Experimental Knowledge*. The University of Chicago Press, Chicago.
- Morgan, M. G. and M. Henrion. (1990). *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge University Press, Cambridge.
- Oberkampf, W.L. and J.C. Helton (2002). Investigation of evidence theory for engineering applications. AIAA Non-Deterministic Approaches Forum, April 2002, Denver, Colorado, paper 2002-1569.

- Oberkampf, W.L., J.C. Helton and K. Sents (2001). Mathematical representation of uncertainty. AIAA Non-Deterministic Approaches Forum, April 2001, Seattle, Washington, paper 2001-1645.
- Plous, S. (1993). *The Psychology of Judgment and Decision Making*. McGraw-Hill, New York.
- Regan, H.M., B.E. Sample and S. Ferson (2002a). Comparison of deterministic and probabilistic calculation of ecological soil screening levels. *Environmental Toxicology and Chemistry* 21: 882-890.
- Regan, H.M., B.K. Hope and S. Ferson (2002b). An analysis and portrayal of uncertainty in a food web exposure model. *Human and Ecological Risk Assessment* 8: 1757-1777.
- Robbins, H.E. (1944). On the measure of a random set, I. *Annals of Mathematical Statistics* 15: 70-74.
- Robbins, H.E. (1945). On the measure of a random set, II. *Annals of Mathematical Statistics* 16: 342-347.
- Saw, J.G., M.C.K. Yang and T.C. Mo (1984). Chebyshev inequality with estimated mean and variance. *The American Statistician* 38: 130-132.
- Saw, J.G., M.C.K. Yang and T.C. Mo (1988). Corrections. *The American Statistician* 42: 166.
- Sents, K. and S. Ferson (2002). *Combination of Evidence in Dempster-Shafer Theory*. SAND2002-0835 Technical Report, Sandia National Laboratories, Albuquerque, NM.
- Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, New Jersey.
- Solana, V. and N.C. Lind (1990). Two principles for data based on probabilistic system analysis. *Proceedings of ICOSSAR '89, 5th International Conferences on Structural Safety and Reliability*. American Society of Civil Engineers, New York.
- Tonon, F., A. Bernardini, and I. Elishakoff. (1999). Concept of random sets as applied to the design of structures and analysis of expert opinions for aircraft crash. *Chaos, Solitons and Fractals* 10: 1855-1868.
- Tonon, F., A. Bernardini, and A. Mammino (2000a). Determination of parameters range in rock engineering by means of Random Sets Theory. *Reliability Engineering and System Safety* 70: 241-261.
- Tonon, F., A. Bernardini, and A. Mammino (2000b). Reliability analysis of rock mass response by means of Random Set Theory. *Reliability Engineering and System Safety* 70:263-282.
- Tonon, F. (2004). On the use of Random Set Theory to bracket the results of Monte Carlo simulations. *Reliable Computing* 10:107-137.
- Tucker, W.T. and S. Ferson (2003). *Probability bounds analysis in environmental risk assessments*. Available at <http://www.ramas.com/pbawhite.pdf>.
- Yager, R.R. (1986). Arithmetic and other operations on Dempster-Shafer structures. *International Journal of Man-machine Studies* 25: 357-366.

External Distribution

Charles E. Anderson
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78284-0510

Bilal Ayyub (2)
Department of Civil Engineering
University of Maryland
College Park, MD 20742-3021

Ivo Babuska
TICAM
Mail Code C0200
University of Texas at Austin
Austin, TX 78712-1085

Osman Balci
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

Timothy Barry
Science-policy, Planning and
Evaluation
USEPA Headquarters
MC2174
401 M Street, S.W.
Washington D.C. 20460

J. Michael Barton
HQ Army Develop. Test Command
Technology Management Div., Rm 245
314 Longs Corner Road
Aberdeen Proving Ground, MD 21005-
5005

Steven Batill (2)
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

John Benek
AFRL/VAAC
2210 Eighth St.
Wright-Patterson AFB, OH 45433

James Berger
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

Daniel Berleant (2)
Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50014

Vicki M. Bier
Department of Industrial Engineering
University of Wisconsin
Madison, WI 53706

Jay Boris (2)
Laboratory for Computational Physics
and Fluid Dynamics
Naval Research Laboratory
Code 6400
4555 Overlook Ave, SW
Washington, DC 20375-5344

Tatiana Borisova
Division of Resource Management
West Virginia University
PO Box 6108
Morgantown, WV 26506-6108

Dima Burmistrov
Menzie-Cura & Associates, Inc.
8 Winchester Place, Suite 202
Winchester, MA 01890

John A. Cafeo
General Motors R&D Center

Mail Code 480-106-256
30500 Mound Road
Box 9055
Warren, MI 48090-9055

James C. Cavendish
General Motors R&D Center
Mail Code 480-106-359
30500 Mound Road
Box 9055
Warren, MI 48090-9055

Chun-Hung Chen (2)
Department of Systems Engineering &
Operations Research
George Mason University
4400 University Drive, MS 4A6
Fairfax, VA 22030

Wei Chen
Department of Mechanical Engineering
Northwestern University
2145 Sheridan Road, Tech B224
Evanston, IL 60208-3111

Kyeongjae Cho (2)
Dept. of Mechanical Engineering
MC 4040
Stanford University
Stanford, CA 94305-4040

Robert T. Clemen
Fuqua School of Business
Box 90120
Duke University
Durham, NC 27708-0120

Hugh Coleman
Department of Mechanical &
Aero. Engineering
University of Alabama/Huntsville
Huntsville, AL 35899

Raymond Cosner (2)
Boeing-Phantom Works
MC S106-7126
P. O. Box 516
St. Louis, MO 63166-0516

Thomas A. Cruse
AFRL Chief Technologist
1981 Monahan Way
Bldg. 12, Room 107
Wright-Patterson AFB, OH 45433-7132

Alison Cullen
University of Washington
Box 353055
208 Parrington Hall
Seattle, Washington 98195-3055

Tod DeLong
Avatar Environmental, LLC
107 S Church Street
West Chester, PA 19382

Department of Energy (3)
Attn: Kevin Greenough, NA-115
D. Kusnezov, NA-114
Jamilah Soudah, NA-114
Forrestal Building
1000 Independence Ave., SW
Washington, DC 20585

Isaac Elishakoff
Dept. of Mechanical Engineering
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431-0991

Ashley Emery
Dept. of Mechanical Engineering
Box 352600
University of Washington
Seattle, WA 98195-2600

John Evans
Harvard Center for Risk Analysis
718 Huntington Avenue
Boston, MA 02115

Scott Ferson (20)
Applied Biomathematics
100 North Country Road
Setauket, New York 11733-1345

H. Christopher Frey
Department of Civil Engineering
North Carolina State University
Raleigh, NC 27695-7908

Marc Garbey
Dept. of Computer Science
Univ. of Houston
501 Philipp G. Hoffman Hall
Houston, Texas 77204-3010

Roger Ghanem
Dept. of Civil Engineering
Johns Hopkins University
Baltimore, MD 21218

James Glimm (2)
Dept. of Applied Math & Statistics
P138A
State University of New York
Stony Brook, NY 11794-3600

Judy Goldsmith
University of Kentucky
Computer Science
773 Anderson Hall
Lexington, Ky. 40506-0046

Philip Goodrum
Syracuse Research
301 Plainfield Road, Suite 350
Syracuse, NY 13212-2510

Ramana Grandhi (2)
Dept. of Mechanical and Materials
Engineering
3640 Colonel Glenn Hwy.
Dayton, OH 45435-0001

Bernard Grossman (2)
The National Institute of Aerospace
144 Research Drive
Hampton, VA 23666

Raphael Haftka (2)
Dept. of Aerospace and Mechanical
Engineering and Engr. Science
P. O. Box 116250
University of Florida
Gainesville, FL 32611-6250

Yacov Haimen
Department of Systems and Information
Engineering
University of Virginia
P.O. Box 400747
151 Engineer's Way
Charlottesville, VA 22904

Achintya Haldar (2)
Dept. of Civil Engineering
& Engineering Mechanics
University of Arizona
Tucson, AZ 85721

Tim Hasselman
ACTA
2790 Skypark Dr., Suite 310
Torrance, CA 90505-5345

Dale Hattis
CENTED
Clark University
950 Main Street
Worcester, MA 01610

George Hazelrigg
Division of Design, Manufacturing
& Innovation
Room 508N
4201 Wilson Blvd.
Arlington, VA 22230

Richard Hills (2)
Mechanical Engineering Dept.
New Mexico State University
P. O. Box 30001/Dept. 3450
Las Cruces, NM 88003-8001

F. Owen Hoffman (2)
SENES
102 Donner Drive
Oak Ridge, TN 37830

Luc Huyse
Southwest Research Institute
6220 Culebra Road
P. O. Drawer 28510
San Antonio, TX 78284-0510

Mark Inlow
Department of Mathematics
Rose-Hulman Institute of Technology
5500 Wabash Avenue
Terre Haute, IN 47803

Rima Izem
Science and Technology Policy Intern
Board of Mathematical Sciences and
their Applications
500 5th Street, NW
Washington, DC 20001

George Karniadakis (2)
Division of Applied Mathematics
Brown University
192 George St., Box F
Providence, RI 02912

Alan Karr
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

George Klir
Binghamton University
Thomas J. Watson School of Engineering
and Applied Sciences
Engineering Building, T-8
Binghamton, NY 13902-6000

Vladik Kreinovich (2)
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968

John Lehner
Energy Sciences and Technology
Building 475, Mailstop 475C
Brookhaven National Laboratory
Upton, New York 11973-5000

W. K. Liu (2)
Northwestern University
Dept. of Mechanical Engineering
2145 Sheridan Road
Evanston, IL 60108-3111

Weldon A. Lodwick
University of Colorado at Denver
Department of Mathematics
Campus Box 170
P.O. Box 173364
Denver, Colorado 80217-3364

Robert Lust
General Motors, R&D and Planning
MC 480-106-256
30500 Mound Road
Warren, MI 48090-9055

Sankaran Mahadevan (2)
Dept. of Civil &
Environmental Engineering
Vanderbilt University
Box 6077, Station B
Nashville, TN 37235

W. McDonald
NDM Solutions
1420 Aldenham Lane
Reston, VA 20190-3901

Richard McGrath
Weston Solutions
10 Lyman Street
Pittsfield, MA 01201

Gregory McRae (2)
Dept. of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

David Meyers
Marine Sciences Research Center
State University of New York
Stony Brook, NY 11794

Sue Minkoff (2)
Dept. of Mathematics and Statistics
University of Maryland
1000 Hilltop Circle
Baltimore, MD 21250

Max Morris (2)
Department of Statistics
Iowa State University
304A Snedecor-Hall
Ames, IW 50011-1210

Rafi Muhanna
Regional Engineering Program
Georgia Tech
210 Technology Circle

Savannah, GA 31407-3039

NASA/Glenn Research Center (2)
Attn: John Slater, MS 86-7
Chris Steffen, MS 5-11
21000 Brookpark Road
Cleveland, OH 44135

NASA/Langley Research Center (7)
Attn: Michael Hensch, MS 499
Tianshu Liu, MS 238
Jim Luckring, MS 286
Joe Morrison, MS 128
Ahmed Noor, MS 369
Sharon Padula, MS 159
Thomas Zang, MS 449
Hampton, VA 23681-0001

C. Needham
Applied Research Associates, Inc.
4300 San Mateo Blvd., Suite A-220
Albuquerque, NM 87110

A. Needleman
Division of Engineering, Box D
Brown University
Providence, RI 02912

Roger B. Nelsen (2)
Dept. of Mathematical Sciences
Lewis & Clark College
0615 SW Palatine Hill Road
Portland OR 97219-7899

Efstratios Nikolaidis (2)
MIME Dept.
4035 Nitschke Hall
University of Toledo
Toledo, OH 43606-3390

Tinsley Oden (2)
TICAM, Mail Code C0200
University of Texas at Austin

Austin, TX 78712-1085

Michael Ortiz (2)
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Chris Paredis
School of Mechanical Engineering
Georgia Institute of Technology
813 Ferst Drive, MARC Rm. 256
Atlanta, GA 30332-0405

Chris L. Pettit
Aerospace Engineering Dept.
MS-11B
590 Holloway Rd.
Annapolis, MD 21402

Gerald R. Prichard
Principal Systems Analyst
Dynetics, Inc.
1000 Explorer Blvd.
Huntsville, AL 35806

Helen Regan
Biology Department
San Diego State University
5500 Campanile Drive
San Diego, CA 92182-4614

John Renaud (2)
Dept. of Aerospace & Mechanical Engr.
University of Notre Dame
Notre Dame, IN 46556

A. J. Rosakis
Graduate Aeronautical Laboratories
California Institute of Technology
1200 E. California Blvd./MS 105-50
Pasadena, CA 91125

Tim Ross (2)
Dept. of Civil Engineering
University of New Mexico
Albuquerque, NM 87131

Chris Roy (2)
Dept. of Aerospace Engineering
211 Aerospace Engineering Bldg.
Auburn University, AL 36849-5338

Michael Runge
USGS Patuxent Wildlife Research Center
12100 Beech Forest Road
Laurel, MD 20708

J. Sacks
Inst. of Statistics and Decision Science
Duke University
Box 90251
Durham, NC 27708-0251

Sunil Saigal (2)
Carnegie Mellon University
Department of Civil and
Environmental Engineering
Pittsburgh, PA 15213

Len Schwer
Schwer Engineering & Consulting
6122 Aaron Court
Windsor, CA 95492

Teddy Seidenfeld
Department of Philosophy, BH 135J
Carnegie Mellon University
135 Baker Hall
Pittsburgh, PA 15213

E. Sevin
Logicon RDA, Inc.
1782 Kenton Circle
Lyndhurst, OH 44124

Ashok Singhal (2)
CFD Research Corp.
Cummings Research Park
215 Wynn Drive
Huntsville, AL 35805

Paul Slovic
Decision Research
1201 Oak Street
Eugene, OR 97401

Bill Spencer (2)
Dept. of Civil Engineering
and Geological Sciences
University of Notre Dame
Notre Dame, IN 46556-0767

Eugene Stakhiv
U.S. Army Corps of Engineers
Institute for Water Resources
Casey Building
7701 Telegraph Road
Alexandria, VA 22315

D. E. Stevenson (2)
Computer Science Department
Clemson University
442 Edwards Hall, Box 341906
Clemson, SC 29631-1906

Ben Thacker
Southwest Research Institute
6220 Culebra Road
P. O. Drawer 28510
San Antonio, TX 78284-0510

Kimberly Thompson
Harvard School of Public Health
617 Huntington Avenue
Boston, MA 02115

Fulvio Tonon (2)
Geology and Geophysics Department
East Room 719
University of Utah
135 South 1460
Salt Lake City, UT 84112

Robert W. Walters (2)
Aerospace and Ocean Engineering
Virginia Tech
215 Randolph Hall, MS 203
Blacksburg, VA 24061-0203

Bill Warren-Hicks
The Cadmus Group, Inc.
Executive Park, Suite 100
1920 Hwy 54
Durham NC 27713

Leonard Wesley
Intellex Inc.
5932 Killarney Circle
San Jose, CA 95138

Justin Y-T Wu
8540 Colonnade Center Drive, Ste 301
Raleigh, NC 27615

Ronald R. Yager
Machine Intelligence Institute
Iona College
715 Noth Avenue
New Rochelle, NY 10801

Ren-Jye Yang
Ford Research Laboratory
MD2115-SRL
P.O.Box 2053
Dearborn, MI 4812

Simone Youngblood (2)
DOD/DMSO
Technical Director for VV&A
1901 N. Beauregard St., Suite 504
Alexandria, VA 22311

M. A. Zikry
North Carolina State University
Mechanical & Aerospace Engineering
2412 Broughton Hall, Box 7910
Raleigh, NC 27695

Foreign Distribution

Tom Aldenberg
RIVM
PO Box 1
NL-3720 BA Bilthoven
NETHERLANDS

Yakov Ben-Haim (2)
Department of Mechanical Engineering
Technion-Israel Institute of Technology
Haifa 32000
ISRAEL

Guillem Bernat
Department of Computer Science
University of York
Heslington
York YO10 5DD
UNITED KINGDOM

Mark Burgman
School of Botany
University of Melbourne
Victoria, 3010
AUSTRALIA

Roger M. Cooke
Delft Institute of Applied Mathematics
Delft University of Technology
PO Box 5031, 2600 GA, Delft
Mekelweg 4, 2628 CD, Delft
THE NETHERLANDS

Gert de Cooman (2)
Universiteit Gent
Onderzoeksgroep, SYSTeMS
Technologiepark - Zwijnaarde 9
9052 Zwijnaarde
BELGIUM

Thomas Fetz
Universität Innsbruck
Technikerstraße 13
A-6020 Innsbruck
AUSTRIA

David Fox
Department of Civil and
Environmental Engineering
University of Melbourne
Victoria, 3010
AUSTRALIA

Marc J. Goovaerts
Katholieke Universiteit Leuven
Departement Toegepaste Economische
Wetenschappen
Naamsestraat 69
B-3000 Leuven
BELGIUM

Dominique Guyonnet
BRGM-Environnemental Department
3 avenue Claude Guillemin
BP 6009
45060 Orléans Cedex 2
FRANCE

Jim Hall (2)
University of Bristol
Department of Civil Engineering
Queen's Building
University Walk
Bristol BS8 1TR
UNITED KINGDOM

Andy Hart
Wildlife Ecotoxicology Unit
Central Science Laboratory
Sand Hatton
York YO41 1LZ
UNITED KINGDOM

Charles Hirsch (2)
Department of Fluid Mechanics
Vrije Universiteit Brussel
Pleinlaan, 2
B-1050 Brussels
BELGIUM

Igor Kozine (2)
Systems Analysis Department
Risø National Laboratory
P. O. Box 49
DK-4000 Roskilde
DENMARK

Ragnar E. Lofstedt
Social Science and Public Policy
King's College London
Strand Building
London WC2R 2LS
UNITED KINGDOM

D. Moens
K. U. Leuven
Dept. of Mechanical Engineering, Div.
PMA
Kasteelpark Arenberg 41
B - 3001 Heverlee
BELGIUM

Dwayne Moore
Cantox Environmental
411 Roosevelt Avenue, Suite 204
Ottawa Ontario K2A 3X9
CANADA

Michael Oberguggenberger (2)
Universität Innsbruck
Technikerstraße 13
A-6020 Innsbruck
AUSTRIA

Nina Nikolova - Jeliaskova
Institute of Parallel Processing

Bulgarian Academy of Sciences
25a "acad. G. Bonchev" str.
Sofia 1113
BULGARIA

K. Papoulia
Inst. Eng. Seismology & Earthquake
Engineering
P.O. Box 53, Finikas GR-55105
Thessaloniki
GREECE

Dominique Pelletier
Genie Mecanique
Ecole Polytechnique de Montreal
C.P. 6079, Succursale Centre-ville
Montreal, H3C 3A7
CANADA

Willem Roelofs
Central Science Laboratory
Sand Hatton
York YO41 1LZ
UNITED KINGDOM

Vincent Sacksteder
Via Eurialo 28, Int. 13
00181 Rome
ITALY

John W. Stevens
AstraZeneca R&D Chernwood
Clinical Sciences
Bakewell Road, Loughborough
Leics LE11 5RH
UNITED KINGDOM

Lev Utkin
Institute of Statistics
Munich University
Ludwigstr. 33
80539, Munich
GERMANY

**Department of Energy
Laboratories**

Los Alamos National Laboratory (19)
Mail Station 5000
P.O. Box 1663
Los Alamos, NM 87545

Attn: Mark C. Anderson, MS T080
Jane M. Booker, MS P946
Terrence Bott, MS K557
Jerry S. Brock, MS F663
C. Chiu, MS F600
Scott Doebling, MS P946
S. Eisenhower, MS K557
Francois Hemez, MS F699
R. Henninger, MS D413
James Hyman, MS B284
Cliff Joslyn, MS B265
Jonathan Lucero, MS C926
Len Margolin, MS D413
Karen I. Pao, MS B256
James Peery, MS F652
Kari Sentz, MS F600
David Sharp, MS B213
Christine Treml, MS J570
Alyson G. Wilson, MS F600

Argonne National Laboratory (2)

Attn: Paul Hovland
Mike Minkoff
MCS Division
Bldg. 221, Rm. C-236
9700 S. Cass Ave.
Argonne, IL 60439

Lawrence Livermore National
Laboratory (6)
7000 East Ave.
P.O. Box 808
Livermore, CA 94550

Attn: Steven Ashby, MS L-561
John Bolstad, MS L-023
Henry Hsieh, MS L-229
Richard Klein, MS L-023
Roger Logan, MS L-125
Charles Tong, MS L-560

Sandia Internal Distribution

1	MS 0512	2500	T. E. Blejwas	1	MS 9159	8962	M. L. Martinez-Canales
1	MS 0769	4100	D. S. Miyoshi	1	MS 0825	9100	A. C. Ratzel
1	MS 0768	4140	A. L. Camp	1	MS 0825	9110	W. Hermina
1	MS 0757	4142	R. D. Waters	1	MS 0825	9115	W. P. Wolfe
1	MS 0757	4142	G. D. Wyss	1	MS 0836	9116	D. Dobranich
1	MS 0751	6111	L. S. Costin	1	MS 0836	9117	R. O. Griffith
1	MS 0735	6115	S. C. James	1	MS 0836	9117	R. J. Buss
1	MS 0751	6117	R. M. Brannon	1	MS 0847	9120	P. J. Wilson
1	MS 0751	6117	A. F. Fossum	1	MS 0847	9124	J. M. Redmond
1	MS 0708	6214	P. S. Veers	1	MS 0847	9124	R. V. Field
1	MS 1138	6221	G. E. Barr	1	MS 0557	9125	T. J. Baca
1	MS 1138	6221	S. M. DeLand	1	MS 0372	9126	R. A. May
1	MS 1138	6222	P. G. Kaplan	1	MS 0372	9127	J. Jung
1	MS 1138	6223	L. M. Claussen	1	MS 0821	9130	A. L. Thornton
1	MS 1137	6223	G. D. Valdez	1	MS 1135	9132	L. A. Gritz
1	MS 0615	6252	J. A. Cooper	1	MS 1135	9132	J. T. Nakos
1	MS 1395	6821	J. W. Garner	1	MS 0836	9132	C. Romero
1	MS 0778	6851	P. N. Swift	1	MS 1135	9132	S. R. Tieszen
1	MS 0778	6851	B. W. Arnold	3	MS 0828	9133	M. Pilch
1	MS 0776	6852	T. Hadgu	1	MS 0828	9133	K. J. Dowding
1	MS 0776	6852	J. S. Stein	1	MS 0828	9133	A. A. Giunta
1	MS 0776	6853	R. P. Rechard	5	MS 0779	9133	J. C. Helton
1	MS 1399	6853	P. Vaughn	20	MS 0828	9133	W. L. Oberkampf
1	MS 0748	6861	D. G. Robinson	1	MS 0557	9133	T. L. Paez
1	MS 0748	6861	S. P. Burns	1	MS 0828	9133	J. R. Red-Horse
1	MS 0742	6870	D. A. Powers	1	MS 0828	9133	V. J. Romero
1	MS 1379	6957	A. Johnson	1	MS 0828	9133	M. P. Sherman
1	MS 0839	7000	L. A. McNamara	1	MS 0828	9133	A. Urbina
1	MS 9153	8200	D. R. Henson	1	MS 0847	9133	W. R. Witkowski
1	MS 9202	8205	R. M. Zurn	1	MS 1135	9134	S. Heffelfinger
1	MS 9014	8242	A. R. Ortega	1	MS 0384	9140	H. S. Morgan
1	MS 9409	8754	W. A. Kawahara	1	MS 0380	9142	K. F. Alvin
1	MS 9042	8763	E. P. Chen	1	MS 1110	9210	D. E. Womble
1	MS 9042	8763	R. E. Jones	1	MS 0370	9211	S. A. Mitchell
1	MS 9042	8763	P. A. Klein	1	MS 0370	9211	M. S. Eldred
1	MS 9405	8763	R. A. Regueiro	1	MS 0370	9211	L. P. Swiler
1	MS 9042	8774	J. J. Dike	3	MS 0370	9211	T. G. Trucano
1	MS 9409	8775	C. D. Moen	1	MS 0428	12300	D. D. Carlson
1	MS 9159	8962	P. D. Hough	1	MS 0428	12330	T. R. Jones
1	MS 9159	8962	K. R. Long	1	MS 0830	12335	K. V. Diegert
				1	MS 0829	12337	W. C. Moffatt
				1	MS 0829	12337	J. M. Sjulín

1 MS 0829 12337 B. M. Rutherford
1 MS 0829 12337 F. W. Spencer
1 MS 0428 12340 V. J. Johnson
1 MS 0638 12341 D. L. Knirk
1 MS 0638 12341 D. E. Peercy
1 MS 0405 12346 S. E. Camp
1 MS 1030 12870 J. G. Miller
1 MS 1005 15240 R. D. Skocypec
1 MS 1176 15242 D. J. Anderson
1 MS 1176 15242 J. E. Campbell
1 MS 1176 15243 R. M. Cranwell
1 MS 1153 15331 L. C. Sanchez
1 MS 1162 15422 W. H. Rutledge
1 MS 1162 15422 L. W. Young
1 MS 1158 15424 K. V. Chavez
1 MS 9018 8945-1 Central
Technical Files
2 MS 0899 9616 Technical
Library