Spring 5-2021

Applications of Evidence Theory to High-Consequence Systems Safety

Christina Marie Deffenbaugh

Christina Marie Deffenbaugh
Candidate

Mathematics and Statistics
Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Erik Erhardt, Chairperson

Dr. Fletcher Christensen

Dr. Walter Gilmore

APPLICATIONS OF EVIDENCE THEORY TO
HIGH-CONSEQUENCE SYSTEMS SAFETY

by

CHRISTINA MARIE DEFFENBAUGH

B.S. Forensic Science, Oklahoma Christian University, 2013

MASTER'S THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Statistics

The University of New Mexico
Albuquerque, New Mexico
May 2021

Dedication

This Master's thesis is dedicated to two people:

First, the late Dr. Stacey Hendrickson: my mentor and champion. Thank you for believing in me and encouraging me, especially during the moments when I failed to believe in myself. I made it, my friend; and I couldn't have done it without you.

And, secondly, to my high school math teacher Megan Garmane. Ms. Garmane, thank you for listening to that one scrawny, shy kid with great handwriting when she asked to take more math classes. I owe this rewarding course of my life to you.

Acknowledgements

Dr. Erik Erhardt: The difference between a having a thesis a student feels good about and one a student feels *great* about comes down to having an excellent thesis advisor. Dr. Erhardt, I cannot thank you enough for taking me on as an advisee, especially during a time when you were plenty busy. I don't know what I would've done without your ideas, coding acumen, gentle reminders to *keep going*, and your availability to help in any and every way you could. I owe so much of this success (and my retained sanity) to you.

Dr. Fletcher Christensen: I will never stop being grateful for you accepting the role of Committee Member during the eleventh hour of developing this thesis. Your feedback and suggestions were a treasure trove of solid improvement, and I deeply appreciate it.

Dr. Walter Gilmore: Walt, I am thankful for your roles as a Committee Member, a coworker, and a friend. Thank you for your contributions to this work and for your continued support in getting it in shape for publication.

Dr. John Darby: There is no way on earth that this thesis could have been completed without your guidance and mentorship. Thank you for being willing to answer *all* of my questions, both great and small, and for being a proverbial font of helpful resources.

Dr. Li Li: My first thesis advisor, I am so appreciative that you took me on as an advisee and for your support and thoughts while we worked together. You are a paragon of kindness, and I count myself fortunate to have been a student of yours.

Bibiana, Gabby, and Lucy: A graduate student isn't complete without a reliable cohort of friends to complain to. Bibiana and Gabby, thank you for all the writing sessions and your motivation and empathy while we worked on our higher-level degrees together; I'm so grateful for our continuing friendship. Lucy, I truly don't know what good thing I did to deserve you as a friend; thank you for being my most passionate cheerleader, for healing my aching soul, and for comforting me when I despaired that I would *never* graduate. If Constancy and Compassion were love languages, then, darling, you are fluent in both.

Lissy, Audrey, Sam, and Krystal: My Dungeons and Dragons wives – *oh!* How excellent you make my days! Thank you for the creative distraction and unparalleled companionship you brought me during such a difficult, uncertain year. Getting to know all four of you has been the most exquisite honor of my life, and I am looking forward to us (and our OCs) growing old together.

Rhys and Agatha: My tiny, furry, feline honorary thesis committee members. There is something about experiencing both writing a Master's thesis and a pandemic that underscores the importance of walking through life with animal companions. Thank you for staying up with me during late writing nights, for always making me laugh, and for reminding me to undertake every task with ecstatic, wild joy.

APPLICATIONS OF EVIDENCE THEORY TO

HIGH-CONSEQUENCE SYSTEMS SAFETY

by

Christina Marie Deffenbaugh

B.S. Forensic Science, Oklahoma Christian University, 2013

M.S. Statistics, The University of New Mexico, 2021

ABSTRACT

Issues linked to abnormal environments (like high-consequence systems safety, e.g., nuclear weapon components, bridges, apartment buildings, etc.) may have insufficient information to use either classical statistical methods or Bayesian approaches for calculating associated probabilistic risks, so there is often a requirement for another method that can deal with a low-information situation to obtain a risk assessment. Belief/plausibility measures of uncertainty from A. P. Dempster and G. Shafer's Evidence Theory is one such method. This thesis has two goals. First, a brief discussion on belief/plausibility measures as an application of Evidence Theory will familiarize the audience with its history and how it can be applied as a general framework for managing problem spaces with significant epistemic

uncertainty – this introduction will be followed by a series of examples which will build in mathematical rigor to finally reach the level of complexity of the problem of interest. Second, the application involves the puncture of three layers of material by a probe to simulate a common safety issue with high-consequence system components – specifically, *what is the minimum energy needed to penetrate through all three layers of material with a 0.10 probability of not exceeding that lowest energy?* This thesis shows that a convolution of three non-interactive belief/plausibility intervals can be used to solve the triple-layer puncture problem, and a sample, step-by-step analysis using simulated expert elicitation data is provided to demonstrate that process.

Table of Contents

ABSTRACT vi
Table of Contents viii
Table of Figures ix
Table of Tables x
Introduction 1
  Expository Example 1: Package delivery with focal element non-overlap 1
  Background 7
  Belief/Plausibility measures 8
  Objective 10
Methods 12
  Basic mathematical terminology 12
  Examples 15
    Example 2: Package delivery with focal element overlap 15
    Example 3: Component housed in a two-layer container 20
    Example 4: Three-layer archery problem with evidence elicited from two SMEs... 26
Results 30
  Application 30
Discussion 33
Conclusion 36
References 37
Appendix 39
  A. Application SME table 39
  B. *BeliefConvolution* code input for BPIs 41
  C. R code for interval and BPI/empirical CDF plots 46

Table of Figures

Figure 1: Example 1 focal elements and assigned evidence (no focal element overlap) 5

Figure 2: Example 1 BPI (no focal element overlap) 6

Figure 3: Example 2 focal elements and assigned evidence (focal element overlap) 16

Figure 4: Example 2 BPI for second delivery time example (focal element overlap).......19

Figure 5: Example 3 focal elements and assigned evidence for two-layer example (pre-convolution) 21

Figure 6: Example 3 focal elements and assigned evidence for two-layer example (post-convolution) 24

Figure 7: Example 3 BPI for two-layer example 25

Figure 8: Example 4 focal elements and assigned evidence for two-SME, three-layer archery example (pre-convolution)...28

Figure 9: Application focal elements and assigned evidence for five-SME, three-layer HCS component problem (pre-convolution) 31

Figure 10: Application BPI for five-SME, three-layer HCS component problem 32

Table of Tables

Table 1: Example 1 focal elements, assigned evidence, and BPI (no focal element overlap) 5

Table 2: Example 2 focal elements and assigned evidence (focal element overlap).........15

Table 3: Example 2 focal element sets and BPI (focal element overlap) 16

Table 4: Example 3 focal elements and assigned evidence for two-layer example (pre-convolution) 20

Table 5: Example 3 convolved focal elements and evidence for two-layer example........23

Table 6: Example 3 focal element sets and BPI for two-layer example 23

Table 7: Example 4 focal elements, assigned evidence, SME weight, and normalized evidence for two-SME, three-layer archery example (pre-convolution) 27

Table 8: Application focal elements, assigned evidence, SME weight, and normalized evidence for five-SME, three-layer HCS component problem (pre-convolution) 40

Introduction

Expository Example 1: Package delivery with focal element non-overlap

Consider the following scenario: A package is expected today, and it is of interest to determine the probability that it will arrive before 1300 hours. Data has not been collected on the exact times at which the packages are delivered; consequently, traditional statistical methods may not be used because the **state of knowledge** (**SOK**) is insufficient to support the assignment of a probability distribution (Ruspini, 1986). However, the time ranges of delivery and a general understanding of how often packages are received during those intervals are known via previous postal service experience at another location, e.g., packages are delivered more often in the afternoon than in the morning. These statements of **expert opinion** regarding the least-to-most frequent package delivery times can be summarized quantitatively by **assigning evidence** to each time interval that there exists an opinion on. For example, as mentioned before, packages are known to be delivered most frequently in the late afternoon; therefore, greater evidence would be assigned to the afternoon time intervals rather than the morning time intervals.

Expert opinion is elicited from **subject matter experts** (**SMEs**), who are individuals considered to be authorities or experts in their professional fields and who can “support informed judgment and prediction about the issues of interest” (Morgan, 2014). Typically, SMEs are trained on the methodology of belief/plausibility measures before they’re asked to gather their collective objective and subjective knowledge on the phenomenon of interest and **assign evidence** (Darby, 2018). The SMEs are asked to proportion evidence that supports the outcome belonging within a range of values – this

"**proportioned evidence**" is weighed information and *not* a probability. When that interval of values has evidence assigned to it, it becomes a **focal element**.

These focal elements and their assigned evidence are collectively used to create **belief/plausibility intervals** (**BPIs**), which are the bounds on the probability of an event occurring (Soundappan et al., 2004). The belief and plausibility bounds are conditioned and updated upon the addition of every piece of evidence that is introduced. **Belief** is the sum of all evidence contained within the set of selected outcomes (lower bound), while **plausibility** is the sum of all evidence that contains and overlaps the set of selected outcomes (upper bound).

The belief bound increases as though all of the evidence on the focal element is a point mass on the upper bound, while the plausibility bound increases as though all of the evidence on the focal element is a point mass on the lower bound. Thus, belief jumps *after the end* of an interval, and plausibility jumps at the *beginning* of an interval. Together they form the BPI, or the lower and upper bounds for a cumulative distribution function (CDF), for any probability distribution that could be assigned to the focal elements. In Table 1 and Figure 1 the mail delivery time intervals are given and have been assigned evidence according to the recipient's SOK based on their previous postal service experience. This is all the information required to create the BPI for this example:

- 0% of the recipient's evidence is assigned to the time range [0000, 0800) – this can be interpreted as, "The recipient, in their experience with their postal service, has no reason to believe that their package will arrive between midnight and 0800". To calculate the BPI for an interval, all of the evidence that has been assigned to other intervals that are contained in or overlap the time before the

earliest time in the interval that we're looking at is evaluated; for example, for Interval 1 (Table 1), we evaluate any evidence assigned before or at midnight. . Plausibility, in terms of Evidence Theory, is a "loose" measure of probability and is defined as the sum of the evidence that totally supports or overlaps to any extent the event of interest (Shafer, 1976), i.e., the sum of the evidence that overlaps or is fully contained in the range that we're evaluating; it is also the upper bound for the BPI. The interval [0000, 0800) overlaps the range of less than or equal to midnight; therefore, the 0.0 evidence is added to the plausibility value. Plausibility = <u>+ 0.0</u> = 0. Belief, on the other hand, is a "strong" or conservative measure of probability and is defined as the sum of only the evidence that totally supports the event of interest (Shafer, 1976), i.e., the sum of the evidence that is fully contained within the range that is being evaluated; it is also the lower bound for the BPI. The interval [0000, 0800) is NOT fully contained under less than or equal to midnight; therefore, the belief remains at its default starting value of zero. This gives us a BPI of [0, 0].

- 10% of the recipient's evidence is assigned to the time range [0800, 1000) – this can be interpreted as, "The recipient, in their experience with their postal service, has little reason to believe that their package will arrive between 0800 and 1000 – a package delivery has happened during this time range before, but it is a rare occurrence". The interval [0800, 1000) overlaps the range of less than or equal to 0800 hours, and the [0000, 0800) interval is fully contained inside it; therefore, the plausibility is the sum of both of these intervals' evidences, which is 0.0 <u>+ 0.1</u> = 0.1. And, now that there is a fully contained interval in this range, evidence can

be added to belief, which in this case, would just be <u>+ 0.0</u> = 0. The BPI for a package arriving at any time in the interval [0800, 1000) is [0, 0.1].

- 10% more of the recipient's evidence is assigned to the time range [1000, 1200). Similarly, Plausibility = 0.0 + 0.1 <u>+ 0.1</u> = 0.2. Belief = 0.0 <u>+ 0.1</u> = 0.1. The BPI for [1000, 1200) is [0.1, 0.2].
- 30% of the evidence is assigned to [1200, 1500). Similarly, Plausibility = 0.0 + 0.1 + 0.1 <u>+ 0.3</u> = 0.5, and Belief = 0.0 + 0.0 + 0.1 <u>+ 0.1</u> = 0.2. The BPI for [1200, 1500) is [0.2, 0.5].
- 40% of the evidence is assigned to [1500, 1800). Plausibility = 0.0 + 0.1 + 0.1 + 0.3 <u>+ 0.4</u> = 0.9, and Belief = 0.0 + 0.0 + 0.1 + 0.1 <u>+ 0.3</u> = 0.5. The BPI for [1500, 1800) is [0.5, 0.9].
- The remaining amount of evidence, 10%, is assigned to [1800, 2000). Plausibility = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 <u>+ 0.1</u> = 1, and Belief = 0.0 + 0.0 + 0.1 + 0.1 + 0.3 <u>+ 0.4</u> = 0.9. The BPI for [1800, 2000) is [0.9, 1].
- 0% evidence is assigned to [2000, 0000). Plausibility = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 + 0.1 <u>+ 0.0</u> = 1, and Belief = 0.0 + 0.0 + 0.1 + 0.1 + 0.3 + 0.4 <u>+ 0.1</u> = 1. The BPI for [2000, 0000) is [1, 1].

The completed BPIs are shown in Figure 2, and can be used to answer the recipient's question: Given the expert opinion provided, what the probability that the package will arrive before 1300? The BPI indicates that the CDF bounds at 1300 are [0.2, 0.5]. Furthermore, between the BPI for this example are three probability distribution CDFs (the blue, red, and green curves) that have parameters consistent with the evidence assigned. This illustrates a few examples of the general principle that no matter what

(sensible) choices we make about the probability distributions defined on the focal elements provided by an SME, the BPIs will always represent the lowest and highest extremes of potential CDF curves.

Table 1: Example 1 focal elements, assigned evidence, and BPI (no focal element overlap)

Interval (Focal Element)	Evidence	BPI
1. [0000, 0800)	0.0	[0.0, 0.0]
2. [0800, 1000)	0.1	[0.0, 0.1]
3. [1000, 1200)	0.1	[0.1, 0.2]
4. [1200, 1500)	0.3	[0.2, 0.5]
5. [1500, 1800)	0.4	[0.5, 0.9]
6. [1800, 2000)	0.1	[0.9, 1.0]
7. [2000, 0000)	0.0	[1.0, 1.0]

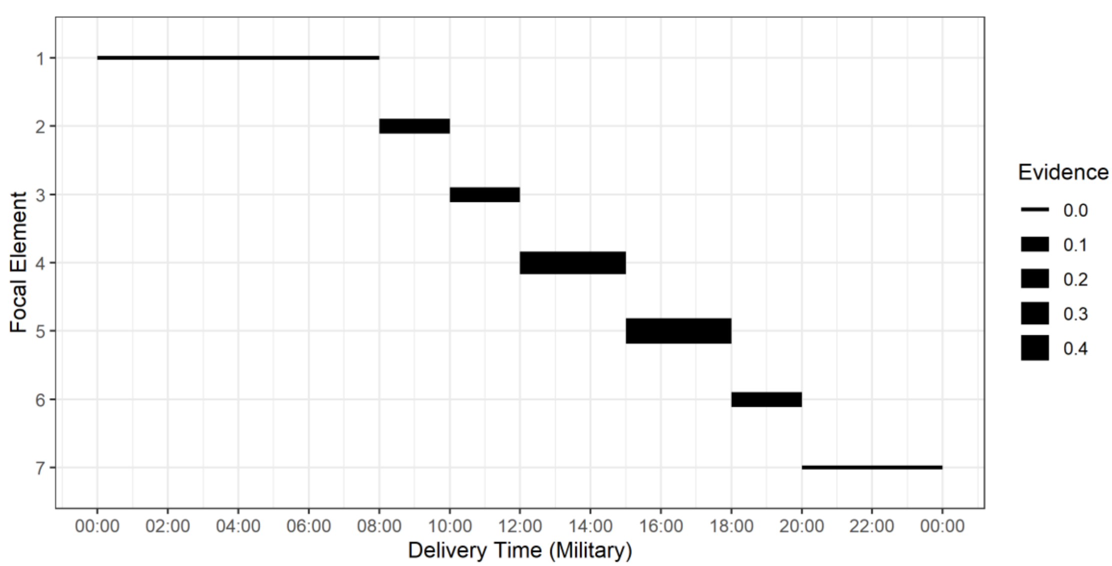

Figure 1: Example 1 focal elements and assigned evidence (no focal element overlap)

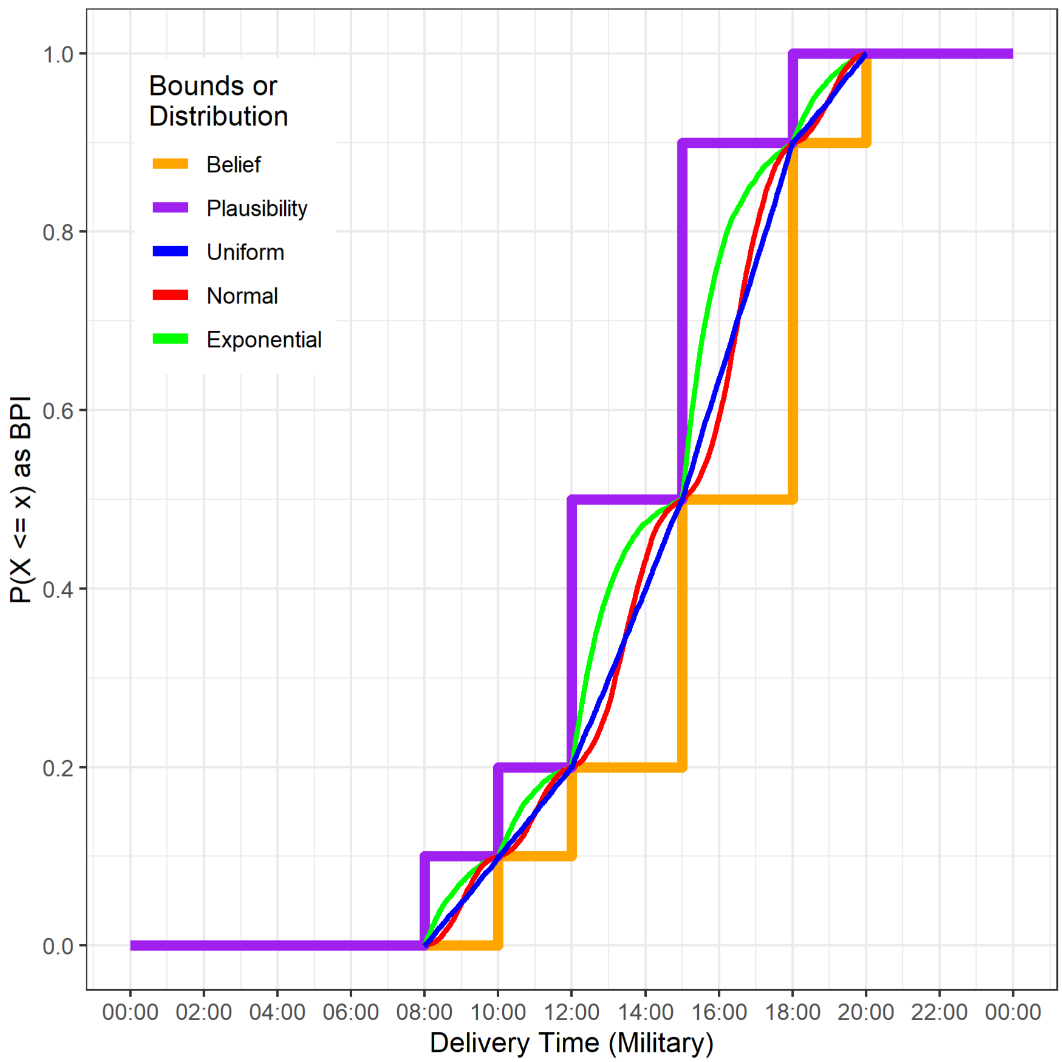

Figure 2: Example 1 BPI (no focal element overlap)

Background

"We must reason and act in an uncertain world" (Friedman & Halpern, 2013). For issues involving **high-consequence systems** (**HCS**) component safety, one value of concern is the probability of an undesirable event, i.e., any phenomenon that affects the usability or safety of the component as a whole.

Undesirable events often occur under conditions that are considered to be abnormal for the system; as a result, databases that track probabilities related to these events are appropriately lacking in content. This makes it difficult to confidently claim that a probability estimate drawn from the database is entirely reliable. Any decisions made based on this limited data are at risk of being ill-informed since large uncertainties limit the integrity of the findings. For this reason, it's extremely challenging to calculate probabilities for these kinds of events (Doorn & Hansson, 2011). To compensate for the lack of data in a real-world setting, controlled experiments are conducted to determine how these components might respond to different environments. But, because of the small sample size – due to the destructive and costly nature of these tests – study results are difficult to interpret with any degree of meaningful precision. Negligible sample sizes diminish statistical power – as such, few inferences, if any, can be reliably drawn from the study.

With enough data, it would be possible to calculate a probability using classical inferential statistics. Even with a moderate amount of data, Bayesian analysis methods could also be used. But, with a sparse data set as is so often encountered in HCS component data, these techniques are not suitable, and "the need for an appropriate representation of uncertainty as a part of an analysis that supports an important decision is almost universally recognized" (Helton & Brooks, 2019). For these special

occurrences, an approach based on Evidence Theory and expert judgment/elicitation is recommended: Belief/plausibility measures of uncertainty (Shafer, 1976).

Belief/Plausibility measures

Belief/Plausibility measures of uncertainty are an expansion of Probability Theory, which are alternatively known as either Evidence Theory or Dempster-Shafer Theory (DST). Historically, this methodology was developed as a way to combine evidence from SMEs when there was a lack of empirical data and to use that information to calculate probability intervals for events of concern. The general framework for this practice of reasoning with uncertainty was first introduced by Arthur P. Dempster in 1967 in the context of statistical inference (Dempster, 1967, 2008), and it was formalized by Glenn Shafer in 1976 as a theory of evidence (Shafer, 1976). Later on, in the 1990s, Philippe Smets further elaborated on these theories and developed the Transferable Belief Model (Klawonn & Smets, 1992; Smets, 1990), which is a framework used to quantify beliefs based on belief functions (Smets & Kennes, 2008). Since then, applications involving belief/plausibility measures have ranged from information fusion to data classification (Chen et al., 2014) to reliability and risk analysis (Denoeux & Masson, 2012).

Uncertainties surrounding these undesirable events can be classified in one of two ways: (1) **aleatory**, i.e., “inherent randomness in the future performance of the system” (Helton & Brooks, 2019)) and (2) **epistemic**, i.e., “uncertainty that arises from a lack of knowledge about the appropriate value to use for an input to the analysis that has a fixed but poorly known value” (Helton & Brooks, 2019). This thesis is interested in the latter, as it has become increasingly evident that “the use of [classical probabilistic methods] to represent epistemic uncertainty can lead to a characterization of epistemic uncertainty

that implies a greater level of knowledge about the uncertainty being represented than is really the case" (Helton & Brooks, 2019).

DST was developed to capture and manage epistemic uncertainty. To better appreciate what this means, it's important to distinguish epistemic uncertainty from aleatory uncertainty. The difference between these two terms can be summarized using the classic coin toss example (Darby, 2018). Suppose that a coin is tossed, and it is known to be a "fair" coin, meaning that the chance of the coin landing heads-up is equal to the chance that the coin lands tails-up. The uncertainty of the coin toss outcome is aleatory, meaning that the chance of either outcome is quantifiable, i.e., the outcome of obtaining a heads-up toss is equal to obtaining a tails-up toss, which is 0.50. The uncertainty of the outcome is due to the randomness of the toss alone. Now say a coin is given, but the trustworthiness of this coin is uncertain – for example, the coin is two-sided, the coin is curved along its diameter, etc. The uncertainty in this situation is now epistemic, meaning that our SOK is inadequate to assign a probability to either outcome. The most that we could say about the outcome of the coin toss is that the probability of obtaining either a tail or a head is between 0 and 1; a BPI has now been established, where belief is equal to 0 and plausibility is equal to 1. In other words, the probability that heads (or tails) will be the outcome of the coin toss is contained within [0, 1].

It is possible to narrow a BPI as more information about the situation is learned. When epistemic uncertainty is eliminated, belief and plausibility converge to a single value, a probability. For example, in reference to the coin toss example above, say the untrustworthy coin is tossed more than once. If both a head and a tail are recorded during the tosses, it is then known that the coin has two sides – with even more tosses, the

fairness of the coin can be determined, thus, narrowing down the options for a probability of a heads (or tails) outcome to a more limited range of values.

There are several advantages and disadvantages to using belief/plausibility measures to estimate lower and upper bounds on the probability for the outcomes of a sample space. One benefit of using this methodology is that "no *a priori* knowledge and no data are required, making it potentially suitable for [quantifying] previously unseen information. Furthermore, a value for ignorance can be expressed, providing information on the uncertainty of a situation" (Chen et al., 2014). In significant epistemic uncertainty situations, this is highly useful, especially since other approaches (e.g., Bayesian inference) require *a priori* knowledge and some data to develop informative models – for example, it would be helpful to use a method that can deal with empirical ignorance when the goal was to predict network attacks in computer systems that had previously never been encountered before. Chen also identifies two disadvantages to using belief/plausibility measures: (1) Calculating BPIs can very quickly become a computationally complex issue, and (2) DST is vulnerable to the "conflicting beliefs management problem" (Chen et al., 2014), which is defined as the issue that occurs when conflicting evidence is gathered from two or more different information sources, and, therefore, can produce contradictory results (Zadeh, 1986). A few methods have been proposed to mitigate the concerns that these two issues pose, but none have been standardized (Chen et al., 2014; Murphy, 2000; Yager, 1987).

Objective

For this thesis, belief/plausibility measures will be used to evaluate the safety of an HCS component. The component is housed under three containers, and exposure of the component via the rupturing of all of the protective layers would cause the component

to be inoperable. Experiments to test the durability of the protective layers would be destructive and costly, which is why data around this phenomenon are difficult to collect. However, SMEs who have years of experience on the component and issues related to its safety are available and willing to contribute their knowledge. This information will be collected during interviews with the SMEs and summarized using belief/plausibility measures to answer the question: *what is the minimum energy needed to penetrate through* <u>*three*</u> *layers of material surrounding the component with a probability of not exceeding that minimum energy being 0.10*?

Methods

Basic mathematical terminology

An understanding of DST starts with a sample space, S, which is a "finite set of mutually exclusive propositions and hypotheses (alternatives) about some problem domain, i.e., the set of all states under consideration" (Chen et al., 2014). Probability is a mapping of all outcomes of a sample space, S, to a value between 0 and 1, i.e., $P: S \rightarrow [0,1]$. The power set of S, $Pow(S)$, is the set of all subsets of S. If S has Θ subsets, $Pow(S)$ has 2^{Θ} elements. For a continuous case, $Pow(S)$ is infinite as there are an infinite number of possible elements, or intervals, to have evidence assigned to.

In Evidence Theory, a mass value, m, is proportioned to elements of $Pow(S)$; in terms of Evidence Theory, m is a value between 0 and 1, and it is the evidence that is assigned to an element of $Pow(S)$,

$$m: 2^{\Theta} \rightarrow [0, 1].$$

Equation 1

The mass function exists when the following two axioms have been upheld (Chen et al., 2014),

1. $m(\emptyset) = 0$, and
2. $\sum_{A \subseteq \Theta} m(A) = 1$.

Equation 2

The first axiom states that the evidence of the null set must be 0, and the second states that the sum of all assigned evidence must equal 1. When an element of $Pow(S)$ is assigned evidence, it is called a focal element (Darby, 2018). The assigned evidence and

their corresponding focal elements are everything needed to calculate the lower and upper bounds of a cumulative probability interval: belief and plausibility, respectively.

Belief, in terms of Evidence Theory, is a “strong” measure of probability and is defined as the sum of only the evidence that totally supports the event of interest, i.e., the sum of all the assigned evidence in that interval; it is also the lower bound for probability. In mathematical terms (Chen et al., 2014), belief is

$$Bel(A) = \sum_{B|B\subseteq A} m(B).$$

Equation 3

$m(B)$ “expresses the proportion of available evidence that supports the claim that the actual state belongs to *B* but not to any subset of *B*” (Chen et al., 2014).

Plausibility, on the other hand, is a “loose” measure of probability and is defined as the sum of the evidence that does not contradict (Helton & Brooks, 2019) the event of interest, i.e., the sum of all the assigned evidence in or overlapping in that interval; it is also the upper bound for probability. Mathematically (Chen et al., 2014), it is defined as

$$Pl(A) = \sum_{B|A\cap B\neq 0} m(B).$$

Equation 4

All in all, these values form a probability interval where

$$m(A) \leq Bel(A) \leq Pl(A).$$

Equation 5

The mathematics supporting belief/measures is relatively straightforward: focal elements and their assigned evidence are all that are required to develop the BPI that

informs solutions to problems dealing with significant epistemic uncertainty. However, the more outcomes that are possible in the problem space, the more tedious the math becomes. (Recall that 2^{Θ} possible focal elements exist when there are Θ outcomes; this can lead to a lot of focal elements and assigned evidence very quickly.) This has led to the creation of computer codes that are able to adeptly handle problems that include a large amount of evidence. But what if the problem includes two or more independent – or **non-interactive** – BPIs that need to be convolved? What if more than one SME provide assigned evidence? The needs of the experiment may differ, but the BPI can be calculated for any of these instances. Several examples that build in mathematical rigor will be presented in order to show how developing a BPI in a variety of experimental conditions can be accomplished. These methods will then be used to solve the problem of interest, which includes evidence assigned by multiple SMEs and a need to convolve more than one BPI.

Examples

Example 2: Package delivery with focal element overlap

In Example 1 involving the package delivery time scenario evidence was assigned to intervals of time that did not overlap. This makes the calculation of the BPIs relatively straightforward. When the information from intervals overlap with others, the information within that overlap needs to be taken into account when calculating the BPIs. This can be done using the definitions of belief and plausibility. Recall that belief is a “strong” measure of probability and plausibility is a “loose” measure. Since plausibility takes into account all of the evidence, including the evidence of the interval being evaluated at that moment, its values tend to “jump” at the *beginnings* of intervals; and since belief only takes into account intervals that have been completely evaluated, its value “jumps” *after the end* of interval has been reached. When intervals overlap, it can be tricky to keep track of this logic to create the resulting BPI. In this example, we have modified the focal elements from Example 1 to overlap. Breaking down the supplied focal elements in Table 2 and Figure 3 into **sets** in as done in Table 3 can help keep track of the calculations.

Focal Element, B_i	**Evidence, $m(B_i)$**
1. [0000, 0800)	0.0
2. [0800, 1030)	0.1
3. [0900, 1300)	0.1
4. [1100, 1530)	0.3
5. [1430, 1830)	0.4
6. [1700, 2000)	0.1
7. [2000, 0000)	0.0

Table 2: Example 2 focal elements and assigned evidence (focal element overlap)

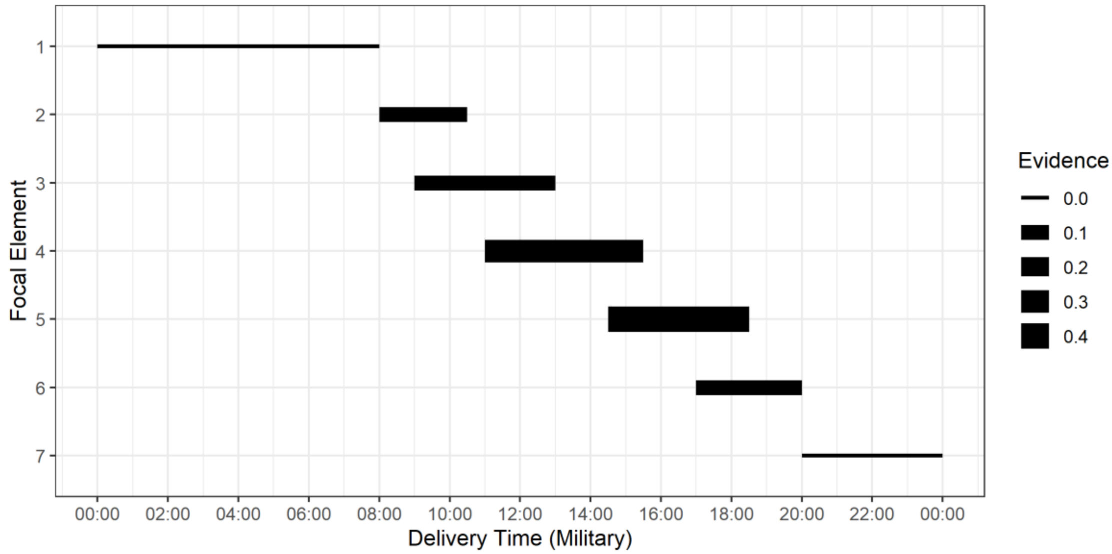

Figure 3: Example 2 focal elements and assigned evidence (focal element overlap)

Set, A_i	**BPI**
1. [0000, 0800)	[0.0, 0.0]
2. [0800, 0900)	[0.0, 0.1]
3. [0900, 1030)	[0.0, 0.2]
4. [1030, 1100)	[0.1, 0.2]
5. [1100, 1300)	[0.1, 0.5]
6. [1300, 1430)	[0.2, 0.5]
7. [1430, 1530)	[0.2, 0.9]
8. [1530, 1700)	[0.5, 0.9]
9. [1700, 1830)	[0.5, 1.0]
10. [1830, 2000)	[0.9, 1.0]
11. [2000, 0000)	[1.0, 1.0]

Table 3: Example 2 focal element sets and BPI (focal element overlap)

Equation 3 and Equation 4 will now be used with Table 3 to calculate the BPI shown in Figure 4 with the given evidence:

- For set A_1 = [0000, 0800), or the belief/plausibility that the package delivery time is < 0800:
    - $Bel(A_1) = \sum_{B|B\subseteq A} m(B) = 0$
    - $Pl(A_1) = \sum_{B|A\cap B\neq 0} m(B) = m(B_1) = 0.0 = 0$
- For set A_2 = [0800, 0900), or the belief/plausibility that the package delivery time is < 0900:
    - $Bel(A_2) = \sum_{B|B\subseteq A} m(B) = m(B_1) = 0.0 = 0$
    - $Pl(A_2) = \sum_{B|A\cap B\neq 0} m(B) = m(B_1) + m(B_2) = 0.0 + 0.1 = 0.1$
- For set A_3 = [0900, 1030):
    - $Bel(A_3) = 0.0 = 0$
    - $Pl(A_3) = 0.0 + 0.1 + 0.1 = 0.2$
- For set A_4 = [1030, 1100):
    - $Bel(A_4) = 0.0 + 0.1 = 0.1$
    - $Pl(A_4) = 0.0 + 0.1 + 0.1 = 0.2$
- For set A_5 = [1100, 1300):
    - $Bel(A_5) = 0.0 + 0.1 = 0.1$
    - $Pl(A_5) = 0.0 + 0.1 + 0.1 + 0.3 = 0.5$
- For set A_6 = [1300, 1430):
    - $Bel(A_6) = 0.0 + 0.1 + 0.1 = 0.2$
    - $Pl(A_6) = 0.0 + 0.1 + 0.1 + 0.3 = 0.5$
- For set A_7 = [1430, 1530):

   - $Bel(A_7) = 0.0 + 0.1 + 0.1 = 0.2$
   - $Pl(A_7) = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 = 0.9$
- For set A_8 = [1530, 1700):
   - $Bel(A_8) = 0.0 + 0.1 + 0.1 + 0.3 = 0.5$
   - $Pl(A_8) = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 = 0.9$
- For set A_9 = [1700, 1830):
   - $Bel(A_9) = 0.0 + 0.1 + 0.1 + 0.3 = 0.5$
   - $Pl(A_9) = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 + 0.1 = 1$
- For set A_{10} = [1830, 2000):
   - $Bel(A_{10}) = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 = 0.9$
   - $Pl(A_{10}) = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 + 0.1 = 1$
- For set A_{11} = [2000, 0000):
   - $Bel(A_{11}) = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 + 0.1 = 1$
   - $Pl(A_{11}) = 0.0 + 0.1 + 0.1 + 0.3 + 0.4 + 0.1 + 0 = 1$

Using the updated focal elements and their assigned evidence, the resulting BPI indicates that the CDF bounds at 1300 are (coincidentally) still [0.20, 0.50], i.e., the probability that a package will arrive before 1300 is between 0.20 and 0.50.

Another way to use BPIs to glean useful information about an event would be to use the probabilities along the y-axis to ask what are the extremes of x given the probability of the event occurring. In the context of this example, the question could be something like, “What is the earliest time that a package will have arrived with a probability of 0.95?” At y = 0.95, Plausibility = 1700 and Belief = 2000, so the earliest time for package delivery is 1700.

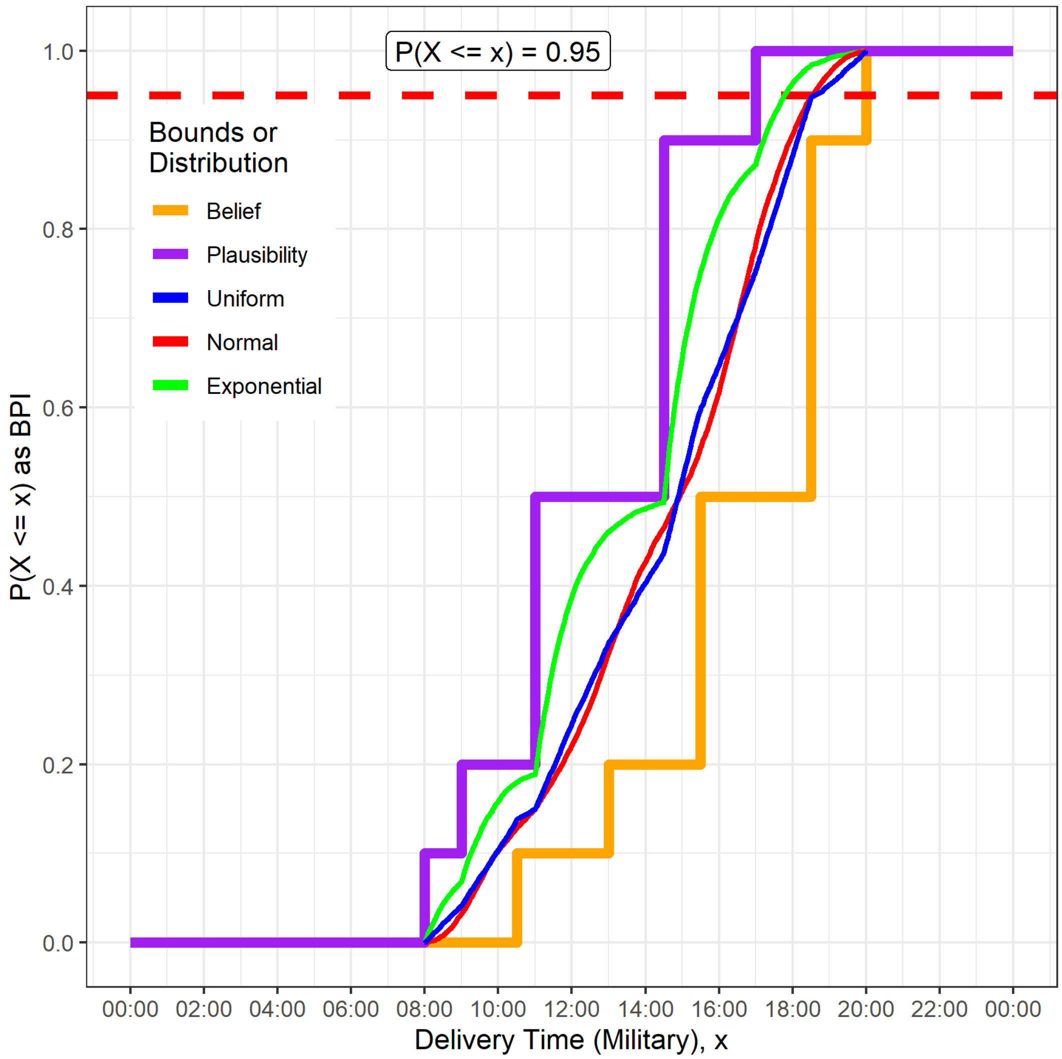

Figure 4: Example 2 BPI for second delivery time example (focal element overlap)

Example 3: Component housed in a two-layer container

The research problem is concerned with the puncture of multiple layers of material using a probe. Adding another barrier complicates the math only slightly such that the evidence must now be convolved with each new barrier that is introduced to the problem space. For example, say that one SME has been hired to assigned evidence to the minimum intervals of energy (energy units are in foot-pound force, ft lbf) needed to puncture through two layers of a given material. First, the SME would need to assign evidence to intervals of energy to puncture the first layer, **E1**, (in the interest of simplifying the mathematics for this problem, only two intervals were assigned evidence), and then assign evidence to energy intervals to puncture the second layer, **E2** (Table 4 and Figure 5. It is assumed that **E1** and **E2** are non-interactive.

Layer	Focal Element, B_i	Evidence, $m(B_i)$
1	1. [500, 600)	0.70
1	2. [420, 700)	0.30
2	3. [400, 800)	0.40
2	4. [480, 700)	0.60

Table 4: Example 3 focal elements and assigned evidence for two-layer example (pre-convolution)

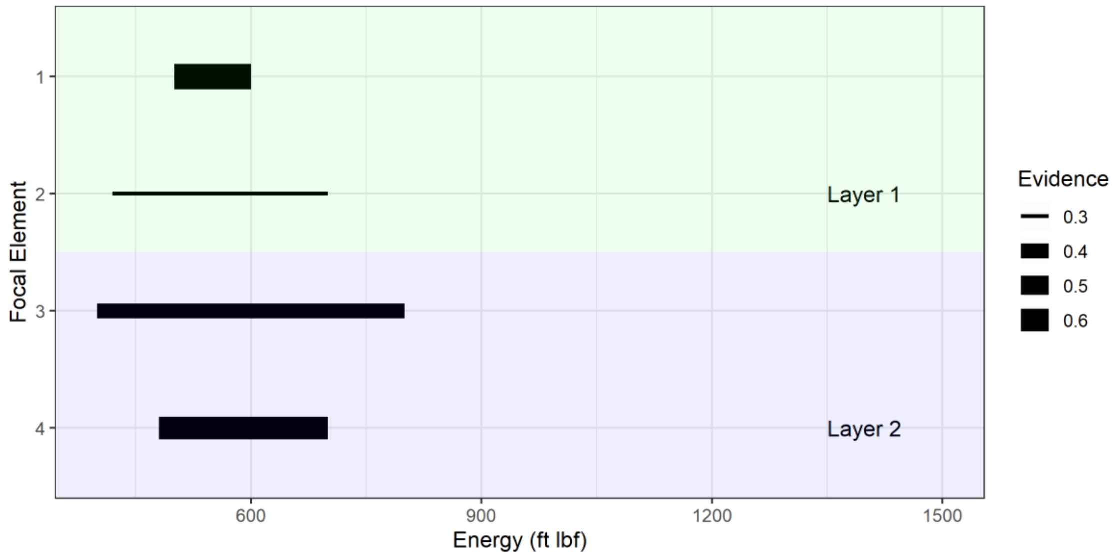

Figure 5: Example 3 focal elements and assigned evidence for two-layer example (pre-convolution)

To find the resulting BPI, the information on the two layers must be convolved by taking their Cartesian product, $f(\mathbf{E1} \times \mathbf{E2})$. $\mathbf{E1}$ and $\mathbf{E2}$ are continuous variables such that $\mathbf{E1} = \{\mathrm{x}: \mathrm{x} \in (0{,}1000)\}$ and $\mathbf{E2} = \{\mathrm{y}: \mathrm{y} \in (0{,}1000)\}$. The function, $f(\mathbf{E1} \times \mathbf{E2})$, is defined as $\{x + y: < x, y > \in \mathbf{E1} \times \mathbf{E2}\}$. f provides the energy that will cause both $\mathbf{E1}$ and $\mathbf{E2}$ to fail, assuming independence. Let $z = f(x, y) = x + y$ denote a specific value for $f(\mathbf{E1} \times \mathbf{E2})$. The intervals over the value of $f(\mathbf{E1} \times \mathbf{E2})$ are:

$\{(z_{min}, z_{max}) | i_{\text{over all } m_{\mathbf{E1}_i}}, j_{\text{over all } m_{\mathbf{E2}_j}}, z_{min} = x_{i_{min}} + y_{j_{min}}, z_{max} = x_{i_{max}} + y_{j_{max}}\}$.

Assuming independence, each interval (z_{min}, z_{max}) has evidence: $m_{(z_{min}, z_{max})} = \sum_* m_{\mathbf{E1}_i} \cdot m_{\mathbf{E2}_j}$ (* all i, j such that $z_{min} = x_{i_{min}} + y_{j_{min}}$ and $z_{max} = x_{i_{max}} + y_{j_{max}}$).

Therefore, for $f(\mathbf{E1} \times \mathbf{E2})$, the resulting convolved evidence is:

$$(\mathbf{500} + \mathbf{400}, \mathbf{600} + \mathbf{800}) = (900, 1400) \quad m = \mathbf{0.70} \cdot \mathbf{0.40} = 0.28$$

$$(\mathbf{500} + \mathbf{480}, \mathbf{600} + \mathbf{700}) = (980, 1300) \quad m = \mathbf{0.70} \cdot \mathbf{0.60} = 0.42$$

$$(\mathbf{420} + \mathbf{400}, \mathbf{700} + \mathbf{800}) = (820, 1500) \quad m = \mathbf{0.30} \cdot \mathbf{0.40} = 0.12$$

$$(\mathbf{420} + \mathbf{480}, \mathbf{700} + \mathbf{700}) = (900, 1400) \quad m = \mathbf{0.30} \cdot \mathbf{0.60} = 0.18$$

Now that the evidence for the two layers of material have been convolved (Table 5 and Figure 6), this information can be used to create the BPI in Table 6 and Figure 7:

- For set A_1 = [0, 820), or the belief/plausibility that the minimum energy required is < 820 ft lbf:
  - $Bel(A_1) = \sum_{B|B\subseteq A} m(B) = 0$
  - $Pl(A_1) = \sum_{B|A\cap B\neq 0} m(B) = 0$
- For set A_2 = [820, 900), or the belief/plausibility that the minimum energy required is < 900 ft lbf:
  - $Bel(A_2) = \sum_{B|B\subseteq A} m(B) = 0$
  - $Pl(A_2) = \sum_{B|A\cap B\neq 0} m(B) = m(B_1) = 0.12$
- For set A_3 = [900, 980):
  - $Bel(A_3) = 0$
  - $Pl(A_3) = 0.12 + 0.28 + 0.18 = 0.58$
- For set A_4 = [980, 1300):
  - $Bel(A_4) = 0$
  - $Pl(A_4) = 0.12 + 0.28 + 0.18 + 0.42 = 1$
- For set A_5 = [1300, 1400):
  - $Bel(A_5) = 0.42 = 0.42$
  - $Pl(A_5) = 0.12 + 0.28 + 0.18 + 0.42 = 1$
- For set A_6 = [1400, 1500):
  - $Bel(A_6) = 0.42 + 0.28 + 0.18 = 0.88$

- $Pl(A_6) = 0.12 + 0.28 + 0.18 + 0.42 = 1$

- For set A_7 = [1500, 2000):
  - $Bel(A_7) = 0.12 + 0.28 + 0.18 + 0.42 = 1$
  - $Pl(A_7) = 0.12 + 0.28 + 0.18 + 0.42 + 0 = 1$

Convolved Focal Element, C_i	**Evidence, $m(C_i)$**
1. [980, 1300)	0.42
2. [820, 1500)	0.12
3. [900, 1400)	0.28
4. [900, 1400)	0.18

Table 5: Example 3 convolved focal elements and evidence for two-layer example

Set, A_i	**BPI**
1. [0000, 0820)	[0.00, 0.00]
2. [0820, 0900)	[0.00, 0.12]
3. [0900, 0980)	[0.00, 0.58]
4. 0[980, 1300)	[0.00, 1.00]
5. [1300, 1400)	[0.42, 1.00]
6. [1400, 1500)	[0.88, 1.00]
7. [1500, 2000)	[1.00, 1.00]

Table 6: Example 3 focal element sets and BPI for two-layer example

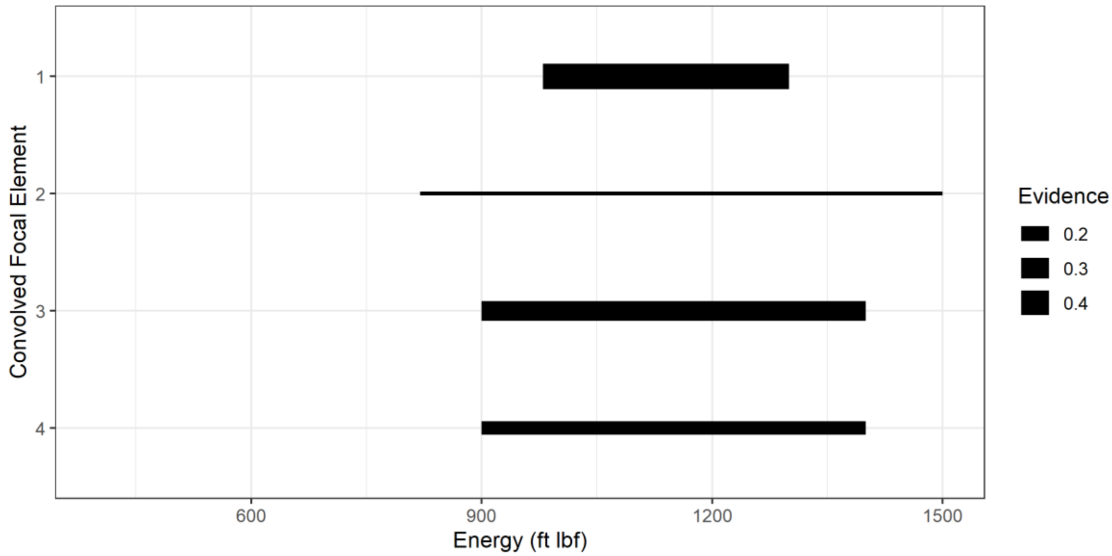

Figure 6: Example 3 focal elements and assigned evidence for two-layer example (post-convolution)

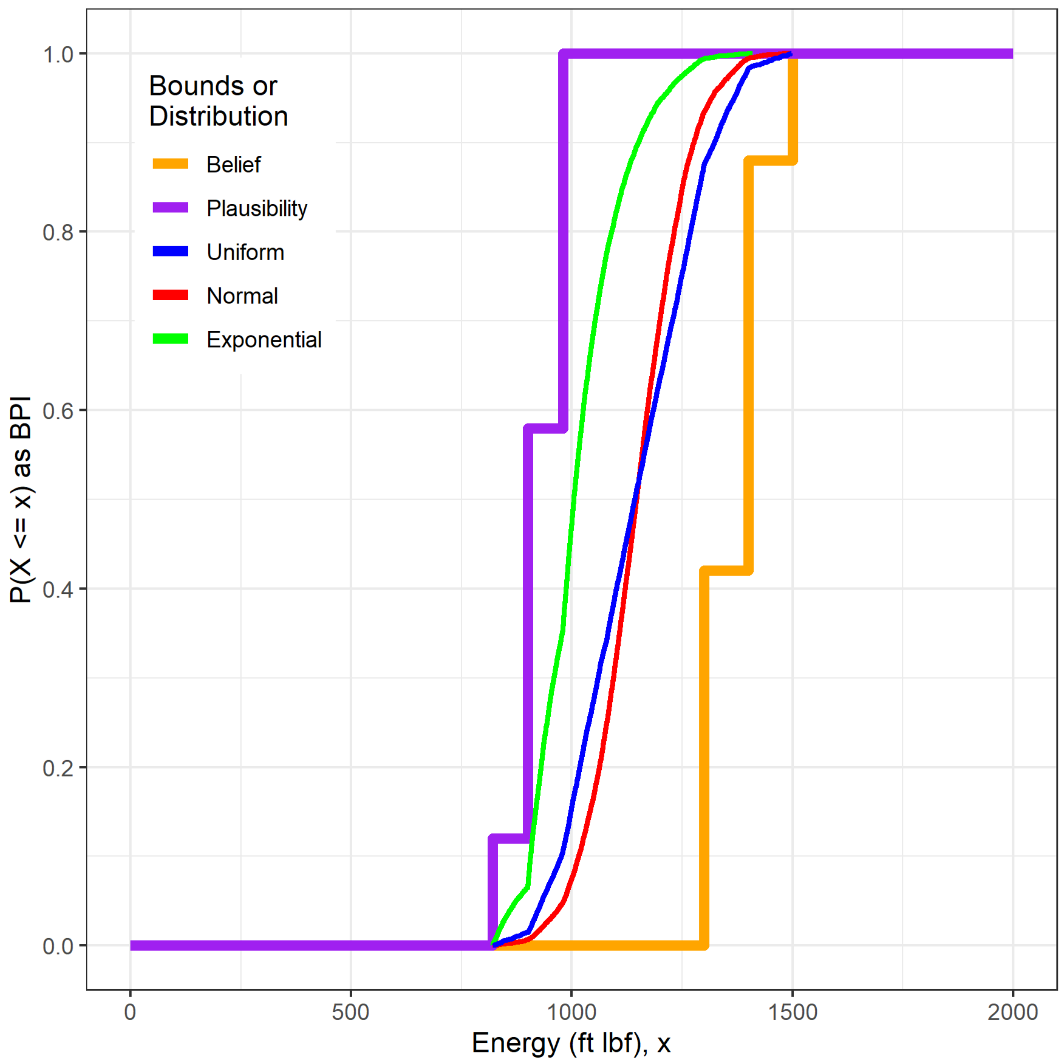

Figure 7: Example 3 BPI for two-layer example

Example 4: Three-layer archery problem with evidence elicited from two SMEs

The next example also requires the convolution of BPIs, but with the addition of more than one SME supplying the evidence. When more than one SME provides expert judgment during the experiment, the evidence they assigned to each focal element is **normalized**. This means that their evidence is weighed according to how reliable each expert is. Usually, each SME's evidence is weighed equally (i.e., $w = \frac{1}{n}$, where n = the number of SMEs), but there are some moments when one SME is known to be more reliable or have more experience with the phenomenon being studied; in this case, that SME's evidence would be weighed more heavily than the others.

The following example is concerned with the minimum amount of strength (SI units) required for an archer to shoot a bow and arrow completely through a three-layered target; if the target is completely pierced through, it will be ruined and require replacement. The layers decrease in density from the outside in, and it is known by the experts that after the first layer (the cardboard stopper) is punctured by the arrow, it does not require much more strength for the arrow to penetrate through the remaining layers. Two SMEs will be assigning evidence during this experiment: SME 1 is an expert on all matters related to archery, both in theory and application, while SME 2 has only moderate experience. Therefore, SME 1's evidence will be weighed three times as much as SME 2's: $w = \frac{3}{4}$ for SME 1 and $w = \frac{1}{4}$ for SME 2. Simply multiply each SME's evidence by their weight to obtain the normalized evidence – this must be done for each separate layer (i.e., for each separate BPI) (Table 7 and Figure 8).

SME	Layer	Focal Element, B_i	Evidence, $m(B_i)$	Weight, w	Normalized Evidence, $m(B_i) \cdot w$
1	1	1. [50, 256)	0.65	0.75	0.4875
1	1	2. [50, 145)	0.35	0.75	0.1875
2	1	3. [150, 256)	0.95	0.25	0.2375
2	1	4. [64, 120)	0.05	0.25	0.0125
1	2	5. [4, 50)	0.10	0.75	0.0750
1	2	6. [16, 32)	0.90	0.75	0.6750
2	2	7. [10, 40)	0.30	0.25	0.0750
2	2	8. [20, 40)	0.70	0.25	0.1750
1	3	9. [1, 3)	0.75	0.75	0.5625
1	3	10. [3, 6)	0.25	0.75	0.1875
2	3	11. [3, 15)	0.45	0.25	0.1125
2	3	12. [5, 16)	0.55	0.25	0.1375

Table 7: Example 4 focal elements, assigned evidence, SME weight, and normalized evidence for two-SME, three-layer archery example (pre-convolution)

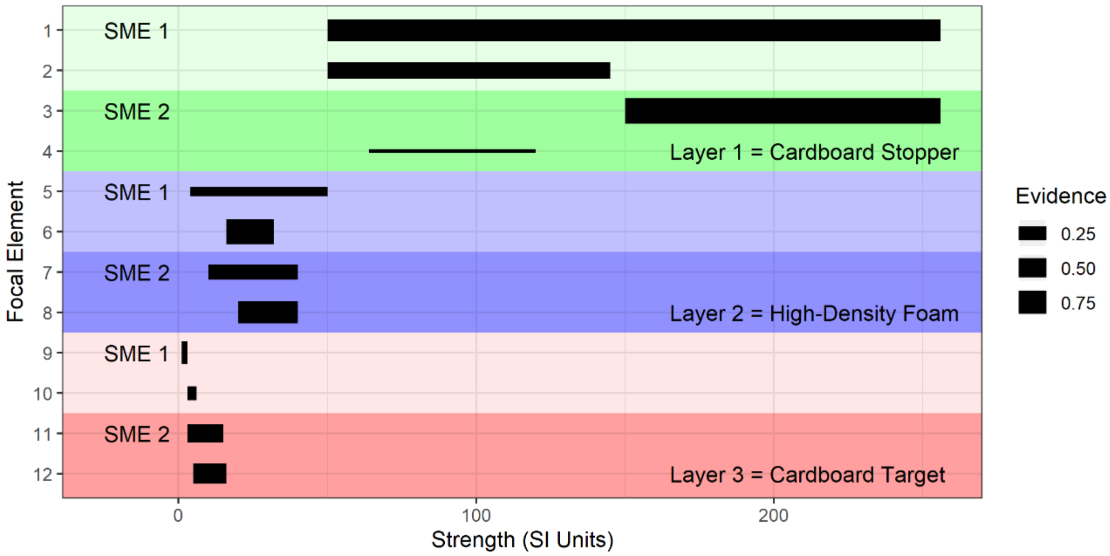

Figure 8: Example 4 focal elements and assigned evidence for two-SME, three-layer archery example (pre-convolution)

At this point, the normalized evidence for each layer would be convolved using a computer program. Such code written in Java has been developed by Dr. John L. Darby at Sandia National Laboratories, and it is known as *BeliefConvolution* (Darby, 2018); it is useful for instances like these where multiple SMEs are used and multiple, independent BPIs need to be convolved – the math, as seen before, is not difficult, but it is tedious, especially as the problem space becomes more complex. (Currently, this software is the property of Sandia National Labs and is not available for distribution and private use.) Since the energy to puncture three, non-interactive layers is being studied, the evidence is convolved under addition operations using the rule for the belief/plausibility measures of uncertainty: This triple sum, **E1** + **E2** + **E3**, is evaluated by first convolving **E1** with **E2**, then convolving that results with **E3**.

In the *BeliefConvolution* code, the three variables are created, and then the variables are assigned "values", i.e., the normalized evidence. Then a variable is created

to hold the result of convolving the first two layers, and the convolution is performed. Similarly, the code creates a variable to hold the result of convolving the first two layers to convolve with the third. So, the code takes the energy variables for the three layers and operates on them two at a time: **E1** convolved (added) with **E2**, then that result convolved (added) with **E3**. The result is the total energy for the failure of all three layers.

So far, we developed a brief guide to belief/plausibility measures of uncertainty and discussed ways to manage problem spaces that increase in complexity – including overlapping focal elements, using multiple BPIs, and engaging multiple SMEs. All of these components will now be used to solve the problem of interest.

Results

Application

Table 8 (in Appendix A), visualized in Figure 9, gives the imitation data set that will be used in solving the three-layer puncture problem. Five simulated SMEs each assigned two focal elements to each of the three layers, and for each focal element assigned evidence. The focal elements are the intervals of energy in foot pound-force exerted on the layers of material. To simplify the structure of this simulation, the energy values provided for the focal elements in the second layer are 18% of the energy estimated to penetrate the first layer, and the third layer values are 13% of the energy estimated to penetrate the first layer, e.g., the strength of Layer 2 for SME 1, [99, 120.6), is 18% of Layer 1 for SME 1, [550, 670). The evidence has been normalized equally between the five SMEs, i.e., $w = \frac{1}{5}$. The analysis of this data will involve inputting all of the data given in Table 8 into the *BeliefConvolution* code, which applies the mathematics used in Example 3, the simple 2-layer problem, with the added function of being able to convolve the 3 BPIs together into one BPI. (The code that was inputted into the *BeliefConvolution* program is included in Appendix B.)

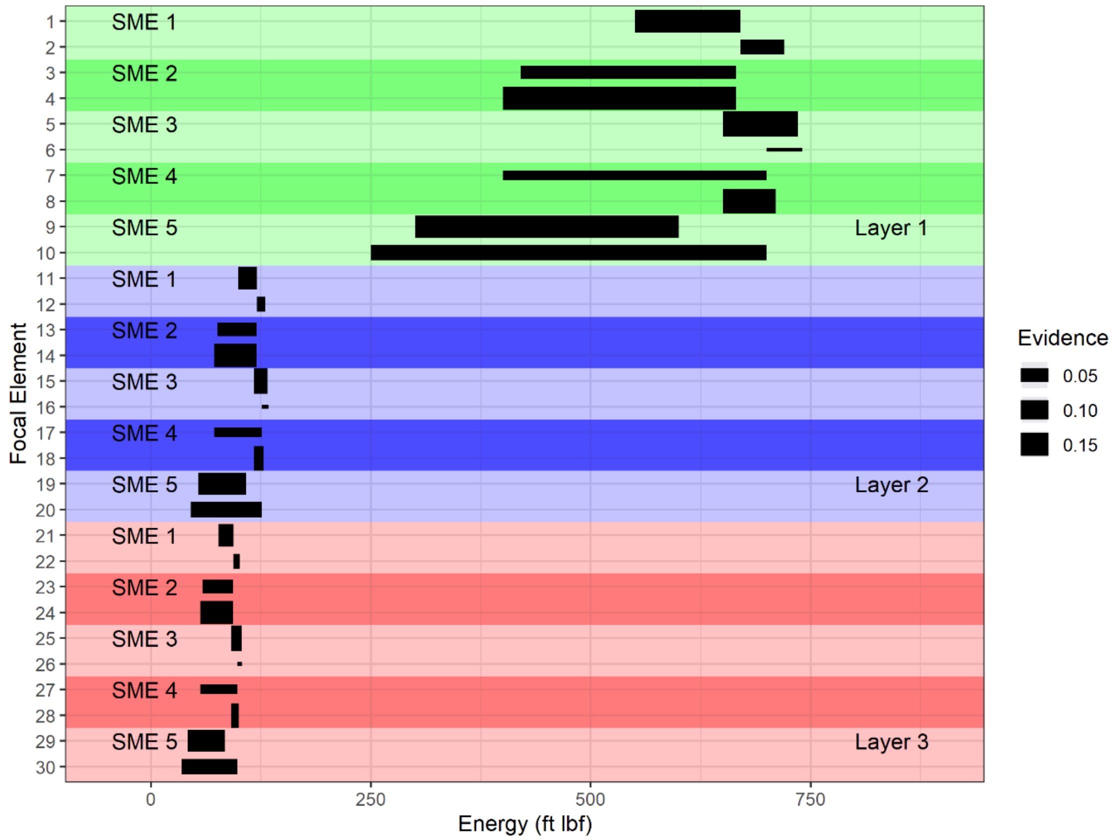

Figure 9: Application focal elements and assigned evidence for five-SME, three-layer HCS component problem (pre-convolution)

The result of this triple convolution is shown in Figure 10. The probability of energy non-exceedance at 0.10 is bounded between 453 and 827 ft lbf; therefore, ***453 ft lbf is the minimum energy*** needed to penetrate through all three layers of material with a 0.10 probability of not exceeding that lowest energy.

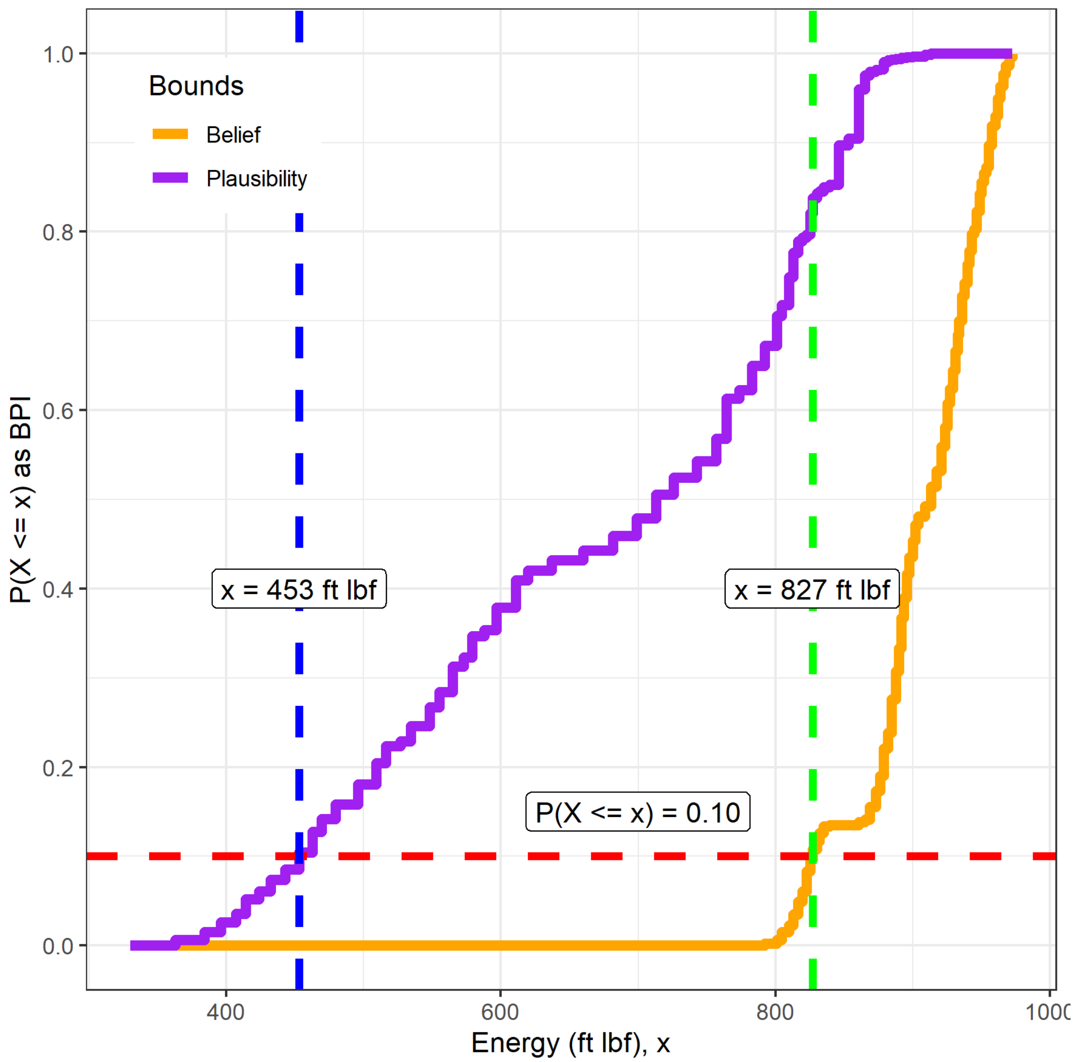

Figure 10: Application BPI for five-SME, three-layer HCS component problem

Discussion

Belief/plausibility measures of uncertainty were used to evaluate the safety of an HCS component without any experimental data. For the hypothetical component in this thesis, 453 ft lbf was determined to be the minimum energy needed to penetrate through all three layers of material with a 0.10 probability of not exceeding that lowest energy; this value would then be used to find the height at which the component may be transported across a facility such that, if the component was dropped, its casing would remain intact and avoid an undesired consequence. Note that the BPI is wide for much of this evaluation, which indicates substantial uncertainty in the results. With large uncertainty, there is significant plausibility that the total energy for failure can be much lower. So, to ensure a very low likelihood of failure, the engineer working with this component could build in a safety factor and select an operating limit value lower than the plausibility limit for the associated energy to puncture, much lower than 453 ft lbf. If this is acceptable, engineered controls would be developed to limit any outside forces to << 453 ft lbf; if this is not acceptable, one of two options remain: (1) Spend the time and money to improve the SOK and provide better evidence, or (2) accept the associated risk.

Methods that attempt to reason through large amounts of uncertainty in a problem space aren't without their limitations. "How to quantify the information-based uncertainty of [evidence]" (Pan et al., 2019) is one such issue. As mentioned before, the conflicting beliefs management problem is another (Chen et al., 2014) – this problem occurs when the evidence assigned to focal elements from two or more SMEs do not agree with each other *and* the evidence has to be convolved, thus creating counterintuitive results. A few new convolution/combination rules have been proposed by Yager and Murphy to minimize this issue, but neither of these has been integrated into

the DST methodology yet (Murphy, 2000; Yager, 1987). Another issue of DST is that it requires subjective assignments of evidence by SMEs; that is, it is based on expert opinion and not objective data. Therefore, this approach suffers from (and benefits from) all issues that are associated with subjectivity. For those concerned with the inherent subjectivity in their results, they may consider using Yager's proposed "credibility qualification" to measure how reliable their SMEs are (Yager, 1987). Despite these challenges, DST remains useful for situations where a decision is required, but the epistemic uncertainty is large, as it provides a structured, axiomatically-defined approach for evaluating such problems.

While the theory around belief/plausibility measures has been well-established, it is relatively new in its application. As mentioned before, belief/plausibility measures have been used for information fusion, data classification, and reliability and risk analysis (Denoeux & Masson, 2012), and many other applications are expected to be explored in the future (Friedman & Halpern, 2013). A potential application would be to use them in the derivation of safety factors. **Safety factors** are the ratios of the average test result to the energy threshold established by expert judgment, and they are often used for systems where comprehensive testing is impractical but require a certain margin of safety against risk. For instance, say that the weight capacity of a bridge is known to be 40,000 pounds – if the engineers designing the bridge applied a safety factor of 2, then the posted limit (the design load), would be $\frac{40{,}000}{2}$ lb = 20,000 lb. The same can be applied to results from belief/plausibility methods. In the case of our application, the lower limit of the BPI would be used to determine the minimum energy applied within the degree of safety: If a safety factor of 2 were also recommended here, the minimum energy applied to the 3-

layers would be $\frac{453}{2}$ ft lbf = 226.5 ft lbf. Given that both safety factors and belief/plausibility measures are used to handle significant epistemic uncertainty, these two methodologies can inform and benefit each other.

Conclusion

In this thesis, belief/plausibility measures, an application of DST, were introduced as a solution for evaluating risk in problem spaces inundated with epistemic uncertainty. One such problem space involves HCS component safety. Belief/plausibility measures were applied to expert-elicited knowledge surrounding a hypothetical situation in which a component was housed within a multilayered casing to find the minimum energy required to puncture through all three layers to reach the component and cause it to fail at probability 0.10. A minimum energy value was determined with this methodology and could be used in the future to inform the maximum height at which the component should be moved. The main contribution was to show how three non-interactive BPIs could be convolved into one BPI and therefore show the probability intervals at which the total energy to penetrate all three barriers is no greater than any specific energy value.

References

Chen, Q., Whitbrook, A., Aickelin, U., & Roadknight, C. (2014). Data classification using the Dempster–Shafer method. *Journal of Experimental & Theoretical Artificial Intelligence*, *26*(4), 493–517.

Darby, J. L. (2018). *Applications of Evidence Theory to Issues with Nuclear Weapons.* Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Dempster, A. P. (1967). Upper and lower probability inferences based on a sample from a finite univariate population. *Biometrika*, *54*(3–4), 515–528.

Dempster, A. P. (2008). Upper and lower probabilities induced by a multivalued mapping. In *Classic works of the Dempster-Shafer theory of belief functions* (pp. 57–72). Springer.

Denoeux, T., & Masson, M. (2012). Belief functions: Theory and applications. *Proceedings of the 2nd International Conference on Belief Functions*, 9–11.

Doorn, N., & Hansson, S. O. (2011). Should probabilistic design replace safety factors? *Philosophy & Technology*, *24*(2), 151–168.

Friedman, N., & Halpern, J. Y. (2013). Plausibility measures: A user's guide. *ArXiv Preprint ArXiv:1302.4947*.

Helton, J. C., & Brooks, D. M. (2019). *Evidence Theory Representations for Loss of Assured Safety in Weak Link/Strong Link Systems.* Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Klawonn, F., & Smets, P. (1992). The dynamic of belief in the transferable belief model and specialization-generalization matrices. *Uncertainty in Artificial Intelligence,* 130–137.

Morgan, M. G. (2014). Use (and abuse) of expert elicitation in support of decision making for public policy. *Proceedings of the National Academy of Sciences*, *111*(20), 7176–7184.

Murphy, C. K. (2000). Combining belief functions when evidence conflicts. *Decision Support Systems*, *29*(1), 1–9.

Pan, Q., Zhou, D., Tang, Y., Li, X., & Huang, J. (2019). A novel belief entropy for measuring uncertainty in dempster-shafer evidence theory framework based on plausibility transformation and weighted hartley entropy. *Entropy*, *21*(2), 163.

Ruspini, E. H. (1986). *The logical foundations of evidential reasoning*. SRI International Menlo Park, CA.

Shafer, G. (1976). *A mathematical theory of evidence* (Vol. 42). Princeton university press.

Smets, P. (1990). The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(5), 447–458.

Smets, P., & Kennes, R. (2008). The transferable belief model. In *Classic Works of the Dempster-Shafer Theory of Belief Functions* (pp. 693–736). Springer.

Soundappan, P., Nikolaidis, E., Haftka, R. T., Grandhi, R., & Canfield, R. (2004). Comparison of evidence theory and Bayesian theory for uncertainty modeling. *Reliability Engineering & System Safety*, *85*(1–3), 295–311.

Yager, R. R. (1987). On the Dempster-Shafer framework and new combination rules. *Information Sciences*, *41*(2), 93–137.

Zadeh, L. A. (1986). A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, *7*(2), 85–85.

Appendix

A. Application SME table

SME	Layer	Focal Element, B_i	Evidence, $m(B_i)$	Weight, w	Normalized Evidence, $m(B_i) \cdot w$
1	1	1. [550, 670]	0.7	0.20	0.14
1	1	2. [670, 720]	0.3	0.20	0.06
2	1	3. [420, 665]	0.25	0.20	0.05
2	1	4. [400, 665]	0.75	0.20	0.15
3	1	5. [650, 735]	0.9	0.20	0.18
3	1	6. [700, 740]	0.1	0.20	0.02
4	1	7. [400, 700]	0.15	0.20	0.03
4	1	8. [650, 710]	0.85	0.20	0.17
5	1	9. [300, 600]	0.675	0.20	0.135
5	1	10. [250, 700]	0.325	0.20	0.065
1	2	11. [99, 120.6]	0.7	0.20	0.14
1	2	12. [120.6, 129.6]	0.3	0.20	0.06
2	2	13. [75.6, 119.7]	0.25	0.20	0.05
2	2	14. [72, 119.7]	0.75	0.20	0.15
3	2	15. [117, 132.3]	0.9	0.20	0.18
3	2	16. [126, 133.2]	0.1	0.20	0.02
4	2	17. [72, 126]	0.15	0.20	0.03
4	2	18. [117, 127.8]	0.85	0.20	0.17
5	2	19. [54, 108]	0.675	0.20	0.135
5	2	20. [45, 126]	0.325	0.20	0.065
1	3	21. [77, 93.8]	0.7	0.20	0.14

1	3	22. [93.8, 100.8]	0.3	0.20	0.06
2	3	23. [58.8, 93.1]	0.25	0.20	0.05
2	3	24. [56, 93.1]	0.75	0.20	0.15
3	3	25. [91, 102.9]	0.9	0.20	0.18
3	3	26. [98, 103.6]	0.1	0.20	0.02
4	3	27. [56, 98]	0.15	0.20	0.03
4	3	28. [91, 99.4]	0.85	0.20	0.17
5	3	29. [42, 84]	0.675	0.20	0.135
5	3	30. [35, 98]	0.325	0.20	0.065

Table 8: Application focal elements, assigned evidence, SME weight, and normalized evidence for five-SME, three-layer HCS component problem (pre-convolution)

B. *BeliefConvolution* code input for BPIs

```
\\ Code Input

Variable minEnergyFail_10_Minus_6_case = new Variable("minEnergyFail_10_Minus_6_case", 0, 1e6, false);


minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 550, 670, 0.7/5)); //
SME_1
minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 670, 720, 0.3/5)); //
SME_1

minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 420, 665, 0.25/5)); //
SME_2
minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 400, 665, 0.75/5)); //
SME_2

minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 650, 735, 0.90/5)); //
SME_3
minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 700, 740, 0.10/5)); //
SME_3

minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 400, 700, 0.15/5)); //
SME_4
minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 650, 710, 0.85/5)); //
SME_4
```

```
minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 300, 600, 0.675/5));
// SME_5
minEnergyFail_10_Minus_6_case.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_case.getName(), 250, 700, 0.325/5));
// SME_5


// #!#!#!#!#!#!#!#!#!  dummy evidence for two components inside case
// assume comp 1 can tolerate 18% of energy that case can tolerate
// assume comp 2 can tolerate 13%  of energy that case can tolerate

Variable minEnergyFail_10_Minus_6_compOne = new Variable("minEnergyFail_10_Minus_6_comp_one", 0, 1e6, false);
Variable minEnergyFail_10_Minus_6_compTwo = new Variable("minEnergyFail_10_Minus_6_comp_two", 0, 1e6, false);

minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 550*0.18,
670*0.18, 0.7/5)); // SME_1
minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 670*0.18,
720*0.18, 0.3/5)); // SME_1

minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 420*0.18,
665*0.18, 0.25/5)); // SME_2
minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 400*0.18,
665*0.18, 0.75/5)); // SME_2
```

```
minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 650*0.18,
735*0.18, 0.90/5)); // SME_3
minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 700*0.18,
740*0.18, 0.10/5)); // SME_3

minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 400*0.18,
700*0.18, 0.15/5)); // SME_4
minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 650*0.18,
710*0.18, 0.85/5)); // SME_4

minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 300*0.18,
600*0.18, 0.675/5)); // SME_5
minEnergyFail_10_Minus_6_compOne.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compOne.getName(), 250*0.18,
700*0.18, 0.325/5)); // SME_5

minEnergyFail_10_Minus_6_compOne.printOverallResults();
minEnergyFail_10_Minus_6_compOne.printExceedanceResultsForExcelPlot();

minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 550*0.14,
670*0.14, 0.7/5)); // SME_1
minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 670*0.14,
720*0.14, 0.3/5)); // SME_1
```

```
minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 420*0.14,
665*0.14, 0.25/5)); // SME_2
minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 400*0.14,
665*0.14, 0.75/5)); // SME_2

minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 650*0.14,
735*0.14, 0.90/5)); // SME_3
minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 700*0.14,
740*0.14, 0.10/5)); // SME_3

minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 400*0.14,
700*0.14, 0.15/5)); // SME_4
minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 650*0.14,
710*0.14, 0.85/5)); // SME_4

minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 300*0.14,
600*0.14, 0.675/5)); // SME_5
minEnergyFail_10_Minus_6_compTwo.addEvidenceInterval(new EvidenceInterval(minEnergyFail_10_Minus_6_compTwo.getName(), 250*0.14,
700*0.14, 0.325/5)); // SME_5


minEnergyFail_10_Minus_6_compTwo.printOverallResults();
minEnergyFail_10_Minus_6_compTwo.printExceedanceResultsForExcelPlot();
```

```
Variable minEnergyFail_10_Minus_6_case_and_compOne =
minEnergyFail_10_Minus_6_case.convoluteNew("minEnergyFail_10_Minus_6__case_and_comp_one", minEnergyFail_10_Minus_6_compOne, "ADD");
Variable minEnergyFail_10_Minus_6_case_and_compOne_and_compTwo =
minEnergyFail_10_Minus_6_case_and_compOne.convoluteNew("minEnergyFail_10_Minus_6__case_and_comp_one_and_comp_two",
minEnergyFail_10_Minus_6_compTwo, "ADD");

minEnergyFail_10_Minus_6_case_and_compOne.printOverallResults();
minEnergyFail_10_Minus_6_case_and_compOne.printExceedanceResultsForExcelPlot();

minEnergyFail_10_Minus_6_case_and_compOne_and_compTwo.printOverallResults();
minEnergyFail_10_Minus_6_case_and_compOne_and_compTwo.printExceedanceResultsForExcelPlot();
```

C. R code for interval and BPI/empirical CDF plots

```
###############
#### Notes ####
###############

# http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/
# Set the 'anchoring point' of the legend (bottom-left is 0, 0; top-right is 1, 1)
# Put bottom-left corner of legend box in bottom-left corner of graph
# Bayesian section: Variable names have hyphens instead of periods


##########################
#### Loading packages ####
##########################

library(ggplot2)
library(reshape2)
library(ggstance)
library(ggpubr)
library(grid)
library(gridExtra)
library(readxl)
library(tidyverse)
library(TruncatedNormal)
library(RGeode)


#######################################################
#### EX: Overlapping = N, SME(s) = 1, Layer(s) = 1 ####
#######################################################

## Plot: Evidence intervals
dat_EI.ON.SME1.L1 <- data.frame(
  `Focal Element` = factor(seq(1:7), levels = c('7', '6', '5', '4', '3', '2', '1'))
  , LB = c(as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 10:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 12:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 15:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 18:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S'))
  , UB = c(as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 10:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 12:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 15:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 18:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 23:59:59', format = '%Y-%m-%d %H:%M:%S'))
  , Evidence = c(0.0, 0.1, 0.1, 0.3, 0.4, 0.1, 0.0))

str(dat_EI.ON.SME1.L1)

dat.long_EI.ON.SME1.L1           <- melt(dat_EI.ON.SME1.L1, measure.vars = c('LB', 'UB'))
colnames(dat.long_EI.ON.SME1.L1) <- c('Focal Element' , 'Evidence', 'Bound', 'Time')

plot_EI.ON.SME1.L1 <- ggplot(
  data = dat.long_EI.ON.SME1.L1, aes(x = Time, y = `Focal Element`, group = Evidence,
size = Evidence)) +
  geom_linerangeh(data = dat.long_EI.ON.SME1.L1, aes(xmin = c(Time[Bound == 'LB'],
Time[Bound == 'LB']), xmax = c(Time[Bound == 'UB'], Time[Bound == 'UB']), y = `Focal
Element`)) +
  scale_x_datetime(breaks = '2 hours', date_labels = '%H:%M') +
  theme_bw() +
  fill_palette('jco') +
  xlab('Delivery Time (Military)')

print(plot_EI.ON.SME1.L1)

ggsave(filename = 'EI.ON.SME1.L1.png', plot = plot_EI.ON.SME1.L1, device = 'png', width =
20, height = 10, units = 'cm')
```

```
## Plot: BP intervals
dat_BPI.ON.SME1.L1 <- data.frame(
  Time = c(as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 10:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 12:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 15:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 18:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 23:59:59', format = '%Y-%m-%d %H:%M:%S'))
  , Belief = c(0, 0, 0.1, 0.2, 0.5, 0.9, 1, 1)
  , Plausibility = c(0, 0.1, 0.2, 0.5, 0.9, 1, 1, 1))

dat.long_BPI.ON.SME1.L1 <- tidyr::pivot_longer(
  data = dat_BPI.ON.SME1.L1
  , cols = c(Belief, Plausibility)
  , names_to = 'Bounds'
  , values_to = 'CDF')

plot_BPI.ON.SME1.L1 <- ggplot(
  dat.long_BPI.ON.SME1.L1, aes(x = Time, y = CDF, colour = Bounds, group = Bounds)) +
  geom_step(size = 2) +
  scale_x_datetime(breaks = '2 hours', date_labels = '%H:%M') +
  scale_color_manual(values = c('orange', 'purple')) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Delivery Time (Military), x') +
  ylab('P(X <= x) as BPI') +
  theme(legend.position = c(0.05, 0.95), legend.justification = c(0, 1))

print(plot_BPI.ON.SME1.L1)

ggsave(filename = 'BPI.ON.SME1.L1.png', plot = plot_BPI.ON.SME1.L1, device = 'png', width
= 20, height = 20, units = 'cm')


#####################################################
#### EX: Overlapping = Y, SME(s) = 1, Layer(s) = 1 ####
#####################################################

## Plot: Evidence intervals
dat_EI.OY.SME1.L1 <- data.frame(
  `Focal Element` = factor(seq(1:7), levels = c('7', '6', '5', '4', '3', '2', '1'))
  , LB = c(as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 09:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 11:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 14:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 17:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S'))
  , UB = c(as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 10:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 13:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 15:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 18:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 23:59:59', format = '%Y-%m-%d %H:%M:%S'))
  , Evidence = c(0.0, 0.1, 0.1, 0.3, 0.4, 0.1, 0.0))

str(dat_EI.OY.SME1.L1)

dat.long_EI.OY.SME1.L1           <- melt(dat_EI.OY.SME1.L1, measure.vars = c('LB', 'UB'))
colnames(dat.long_EI.OY.SME1.L1) <- c('Focal Element' , 'Evidence', 'Bound', 'Time')

plot_EI.OY.SME1.L1 <- ggplot(
  data = dat.long_EI.OY.SME1.L1, aes(x = Time, y = `Focal Element`, group = Evidence,
size = Evidence)) +
  geom_linerangeh(data = dat.long_EI.OY.SME1.L1, aes(xmin = c(Time[Bound == 'LB'],
Time[Bound == 'LB']), xmax = c(Time[Bound == 'UB'], Time[Bound == 'UB']), y = `Focal
Element`)) +
```

```
  scale_x_datetime(breaks = '2 hours', date_labels = '%H:%M') +
  theme_bw() +
  fill_palette('jco') +
  xlab('Delivery Time (Military)')

print(plot_EI.OY.SME1.L1)

ggsave(filename = 'EI.OY.SME1.L1.png', plot = plot_EI.OY.SME1.L1, device = 'png', width =
20, height = 10, units = 'cm')

## Plot: BP intervals
dat_BPI.OY.SME1.L1 <- data.frame(
  Time = c(as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 09:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 10:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 11:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 13:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 14:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 15:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 17:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 18:30:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S')
           , as.POSIXct('2020-01-01 23:59:59', format = '%Y-%m-%d %H:%M:%S'))
  , Belief = c(0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.5, 0.5, 0.9, 1, 1)
  , Plausibility = c(0, 0.1, 0.2, 0.2, 0.5, 0.5, 0.9, 0.9, 1, 1, 1, 1))

dat.long_BPI.OY.SME1.L1 <- tidyr::pivot_longer(
  data = dat_BPI.OY.SME1.L1
  , cols = c(Belief, Plausibility)
  , names_to = 'Bounds'
  , values_to = 'CDF')

plot_BPI.OY.SME1.L1 <- ggplot(
  dat.long_BPI.OY.SME1.L1, aes(x = Time, y = CDF, colour = Bounds, group = Bounds)) +
  geom_step(size = 2) +
  scale_x_datetime(breaks = '2 hours', date_labels = '%H:%M') +
  scale_color_manual(values = c('orange', 'purple')) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Delivery Time (Military), x') +
  ylab('P(X <= x) as BPI') +
  theme(legend.position = c(0.05, 0.95), legend.justification = c(0, 1))

print(plot_BPI.OY.SME1.L1)

ggsave(filename = 'BPI.OY.SME1.L1.png', plot = plot_BPI.OY.SME1.L1, device = 'png', width
= 20, height = 20, units = 'cm')


#######################################################
#### EX: Overlapping = Y, SME(s) = 1, Layer(s) = 2 ####
#######################################################

## Plot: Evidence intervals
# 2-layer, separated interval plot
dat_EI.OY.SME1.L2.A <- data.frame(
  `Focal Element` = factor(seq(1:4), levels = c('4', '3', '2', '1'))
  , LB = c(500, 420, 400, 480)
  , UB = c(600, 700, 800, 700)
  , Evidence = c(0.70, 0.30, 0.40, 0.60))

str(dat_EI.OY.SME1.L2.A)

dat.long_EI.OY.SME1.L2.A           <- melt(dat_EI.OY.SME1.L2.A, measure.vars = c('LB',
'UB'))
colnames(dat.long_EI.OY.SME1.L2.A) <- c('Focal Element' , 'Evidence', 'Bound', 'Energy')

plot_EI.OY.SME1.L2.A <- ggplot(
  data = dat.long_EI.OY.SME1.L2.A, aes(x = Energy, y = `Focal Element`, group = Evidence,
size = Evidence)) +
```

```
  geom_linerangeh(data = dat.long_EI.OY.SME1.L2.A, aes(xmin = c(Energy[Bound == 'LB'],
Energy[Bound == 'LB']), xmax = c(Energy[Bound == 'UB'], Energy[Bound == 'UB']), y =
`Focal Element`)) +
  scale_x_continuous(limits = c(400, 1500)) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Energy (ft lbf)') +
  annotate('text', x = 1350, y = 1, hjust = 0, label = 'Layer 2') +
  annotate('text', x = 1350, y = 3, hjust = 0, label = 'Layer 1') +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 2.5, ymax = Inf), fill = 'green', alpha =
0.007) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = 2.5), fill = 'blue', alpha =
0.007)

ggsave(filename = 'EI.OY.SME1.L2.A.png', plot = plot_EI.OY.SME1.L2.A, device = 'png',
width = 20, height = 10, units = 'cm')

# 2-layer, convoluted interval plot
dat_EI.OY.SME1.L2.B <- data.frame(
  `Focal Element` = factor(seq(1:4), levels = c('4', '3', '2', '1'))
  , LB = c(980, 820, 900, 900)
  , UB = c(1300, 1500, 1400, 1400)
  , Evidence = c(0.42, 0.12, 0.28, 0.18))

str(dat_EI.OY.SME1.L2.B)

dat.long_EI.OY.SME1.L2.B           <- melt(dat_EI.OY.SME1.L2.B, measure.vars = c('LB',
'UB'))
colnames(dat.long_EI.OY.SME1.L2.B) <- c('Focal Element' , 'Evidence', 'Bound', 'Energy')

plot_EI.OY.SME1.L2.B <- ggplot(
  data = dat.long_EI.OY.SME1.L2.B, aes(x = Energy, y = `Focal Element`, group = Evidence,
size = Evidence)) +
  geom_linerangeh(data = dat.long_EI.OY.SME1.L2.B, aes(xmin = c(Energy[Bound == 'LB'],
Energy[Bound == 'LB']), xmax = c(Energy[Bound == 'UB'], Energy[Bound == 'UB']), y =
`Focal Element`)) +
  scale_x_continuous(limits = c(400, 1500)) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Energy (ft lbf)') +
  ylab('Convolved Focal Element')

ggsave(filename = 'EI.OY.SME1.L2.B.png', plot = plot_EI.OY.SME1.L2.B, device = 'png',
width = 20, height = 10, units = 'cm')

## Plot: BP intervals
dat_BPI.OY.SME1.L2 <- data.frame(
  Energy = c(0, 820, 900, 980, 1300, 1400, 1500, 2000)
  , Belief = c(0, 0, 0, 0, 0.42, 0.88, 1, 1)
  , Plausibility = c(0, 0.12, 0.58, 1, 1, 1, 1, 1))

dat.long_BPI.OY.SME1.L2 <- tidyr::pivot_longer(
  data = dat_BPI.OY.SME1.L2
  , cols = c(Belief, Plausibility)
  , names_to = 'Bounds'
  , values_to = 'CDF')

plot_BPI.OY.SME1.L2 <- ggplot(
  dat.long_BPI.OY.SME1.L2, aes(x = Energy, y = CDF, colour = Bounds, group = Bounds)) +
  geom_step(size = 2) +
  scale_color_manual(values = c('orange', 'purple')) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Energy (ft lbf), x') +
  ylab('P(X <= x) as BPI') +
  theme(legend.position = c(0.05, 0.95), legend.justification = c(0, 1))

print(plot_BPI.OY.SME1.L2)

ggsave(filename = 'BPI.OY.SME1.L2.png', plot = plot_BPI.OY.SME1.L2, device = 'png', width
= 20, height = 20, units = 'cm')
```

```
####################################################
#### EX: Overlapping = Y, SME(s) = 2, Layer(s) = 3 ####
####################################################

## Plot: Evidence intervals
dat_EI.OY.SME2.L3 <- data.frame(
  EE = c('SME 1', 'SME 1', 'SME 2', 'SME 2', 'SME 1', 'SME 1', 'SME 2', 'SME 2', 'SME 1',
'SME 1', 'SME 2', 'SME 2')
  , `Focal Element` = factor(seq(1:12), levels = c('12', '11', '10', '9', '8', '7', '6',
'5', '4', '3', '2', '1'))
  , LB = c(50, 50, 150, 64, 4, 16, 10, 20, 1, 3, 3, 5)
  , UB = c(256, 145, 256, 120, 50, 32, 40, 40, 3, 6, 15, 16)
  , Evidence = c(0.65, 0.35, 0.95, 0.05, 0.10, 0.90, 0.30, 0.70, 0.75, 0.25, 0.45, 0.55))

str(dat_EI.OY.SME2.L3)

dat.long_EI.OY.SME2.L3           <- melt(dat_EI.OY.SME2.L3, measure.vars = c('LB', 'UB'))
colnames(dat.long_EI.OY.SME2.L3) <- c('EE', 'Focal Element' , 'Evidence', 'Bound',
'Strength')

plot_EI.OY.SME2.L3 <- ggplot(data = dat.long_EI.OY.SME2.L3, aes(x = Strength, y = `Focal
Element`, group = Evidence, size = Evidence)) +
  geom_linerangeh(data = dat.long_EI.OY.SME2.L3, aes(xmin = c(Strength[Bound == 'LB'],
Strength[Bound == 'LB']), xmax = c(Strength[Bound == 'UB'], Strength[Bound == 'UB']), y =
`Focal Element`)) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Strength (SI Units)') +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = 2.5), fill = 'red', alpha =
0.020) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 2.5, ymax = 4.5), fill = 'red', alpha =
0.005) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 4.5, ymax = 6.5), fill = 'blue', alpha =
0.025) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 6.5, ymax = 8.5), fill = 'blue', alpha =
0.010) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 8.5, ymax = 10.5), fill = 'green', alpha
= 0.020) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 10.5, ymax = Inf), fill = 'green', alpha
= 0.005) +
  annotate('text', x = -25, y = 2, hjust = 0, label = 'SME 2') +
  annotate('text', x = -25, y = 4, hjust = 0, label = 'SME 1') +
  annotate('text', x = -25, y = 6, hjust = 0, label = 'SME 2') +
  annotate('text', x = -25, y = 8, hjust = 0, label = 'SME 1') +
  annotate('text', x = -25, y = 10, hjust = 0, label = 'SME 2') +
  annotate('text', x = -25, y = 12, hjust = 0, label = 'SME 1') +
  annotate('text', x = 165, y = 1, hjust = 0, label = 'Layer 3 = Cardboard Target') +
  annotate('text', x = 165, y = 5, hjust = 0, label = 'Layer 2 = High-Density Foam') +
  annotate('text', x = 165, y = 9, hjust = 0, label = 'Layer 1 = Cardboard Stopper')

plot_EI.OY.SME2.L3 <-
  plot_EI.OY.SME2.L3 +
  geom_linerangeh(data = dat.long_EI.OY.SME2.L3, aes(xmin = c(Strength[Bound == 'LB'],
Strength[Bound == 'LB']), xmax = c(Strength[Bound == 'UB'], Strength[Bound == 'UB']), y =
`Focal Element`))

print(plot_EI.OY.SME2.L3)

ggsave(filename = 'EI.OY.SME2.L3.png', plot = plot_EI.OY.SME2.L3, device = 'png', width =
20, height = 10, units = 'cm')


#########################################################
#### RESULTS: Overlapping = Y, SME(s) = 5, Layer(s) = 3 ####
#########################################################

## Plot: Evidence intervals
dat_EI.OY.SME5.L3 <- read_excel(path = 'Dummy_Data.xlsx', sheet = 1)
dat_EI.OY.SME5.L3 <- dat_EI.OY.SME5.L3[, 2:6]
```

```
colnames(dat_EI.OY.SME5.L3)       <- c('EE', 'Focal Element', 'LB', 'UB', 'Evidence')
dat_EI.OY.SME5.L3                 <- as.data.frame(dat_EI.OY.SME5.L3)
dat_EI.OY.SME5.L3$`Focal Element` <- factor(dat_EI.OY.SME5.L3$`Focal Element`, levels =
c('30', '29', '28', '27', '26', '25', '24', '23', '22', '21', '20', '19', '18', '17',
'16', '15', '14', '13', '12', '11', '10', '9', '8', '7', '6', '5', '4', '3', '2', '1'))
dat_EI.OY.SME5.L3$Evidence        <- as.character(dat_EI.OY.SME5.L3$Evidence)
dat_EI.OY.SME5.L3$Evidence        <- as.numeric(dat_EI.OY.SME5.L3$Evidence)

dat.long_EI.OY.SME5.L3           <- melt(dat_EI.OY.SME5.L3, measure.vars = c('LB', 'UB'))
colnames(dat.long_EI.OY.SME5.L3) <- c('EE', 'Focal Element' , 'Evidence', 'Bound',
'Energy')
dat.long_EI.OY.SME5.L3$Energy    <- as.numeric(dat.long_EI.OY.SME5.L3$Energy)

plot_EI.OY.SME5.L3 <- ggplot(data = dat.long_EI.OY.SME5.L3, aes(x = Energy, y = `Focal
Element`, group = Evidence, size = Evidence)) +
  geom_linerangeh(data = dat.long_EI.OY.SME5.L3, aes(xmin = c(Energy[Bound == 'LB'],
Energy[Bound == 'LB']), xmax = c(Energy[Bound == 'UB'], Energy[Bound == 'UB']), y =
`Focal Element`)) +
  scale_x_continuous(limits = c(-50, 900)) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Energy (ft lbf)') +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = 2.5), fill = 'red', alpha =
0.0025) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 2.5, ymax = 4.5), fill = 'red', alpha =
0.010) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 4.5, ymax = 6.5), fill = 'red', alpha =
0.0025) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 6.5, ymax = 8.5), fill = 'red', alpha =
0.010) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 8.5, ymax = 10.5), fill = 'red', alpha =
0.0025) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 10.5, ymax = 12.5), fill = 'blue', alpha
= 0.005) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 12.5, ymax = 14.5), fill = 'blue', alpha
= 0.020) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 14.5, ymax = 16.5), fill = 'blue', alpha
= 0.005) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 16.5, ymax = 18.5), fill = 'blue', alpha
= 0.020) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 18.5, ymax = 20.5), fill = 'blue', alpha
= 0.005) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 20.5, ymax = 22.5), fill = 'green', alpha
= 0.0025) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 22.5, ymax = 24.5), fill = 'green', alpha
= 0.010) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 24.5, ymax = 26.5), fill = 'green', alpha
= 0.0025) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 26.5, ymax = 28.5), fill = 'green', alpha
= 0.010) +
  geom_rect(aes(xmin = -Inf, xmax = Inf, ymin = 28.5, ymax = Inf), fill = 'green', alpha
= 0.0025) +
  annotate('text', x = -45, y = 2, hjust = 0, label = 'SME 5') +
  annotate('text', x = -45, y = 4, hjust = 0, label = 'SME 4') +
  annotate('text', x = -45, y = 6, hjust = 0, label = 'SME 3') +
  annotate('text', x = -45, y = 8, hjust = 0, label = 'SME 2') +
  annotate('text', x = -45, y = 10, hjust = 0, label = 'SME 1') +
  annotate('text', x = -45, y = 12, hjust = 0, label = 'SME 5') +
  annotate('text', x = -45, y = 14, hjust = 0, label = 'SME 4') +
  annotate('text', x = -45, y = 16, hjust = 0, label = 'SME 3') +
  annotate('text', x = -45, y = 18, hjust = 0, label = 'SME 2') +
  annotate('text', x = -45, y = 20, hjust = 0, label = 'SME 1') +
  annotate('text', x = -45, y = 22, hjust = 0, label = 'SME 5') +
  annotate('text', x = -45, y = 24, hjust = 0, label = 'SME 4') +
  annotate('text', x = -45, y = 26, hjust = 0, label = 'SME 3') +
  annotate('text', x = -45, y = 28, hjust = 0, label = 'SME 2') +
  annotate('text', x = -45, y = 30, hjust = 0, label = 'SME 1') +
  annotate('text', x = 800, y = 2, hjust = 0, label = 'Layer 3') +
  annotate('text', x = 800, y = 12, hjust = 0, label = 'Layer 2') +
  annotate('text', x = 800, y = 22, hjust = 0, label = 'Layer 1')
```

```
plot_EI.OY.SME5.L3 <-
  plot_EI.OY.SME5.L3 +
  geom_linerangeh(data = dat.long_EI.OY.SME5.L3, aes(xmin = c(Energy[Bound == 'LB'],
Energy[Bound == 'LB']), xmax = c(Energy[Bound == 'UB'], Energy[Bound == 'UB']), y =
`Focal Element`))

print(plot_EI.OY.SME5.L3)

ggsave(filename = 'EI.OY.SME5.L3.png', plot = plot_EI.OY.SME5.L3, device = 'png', width =
20, height = 15, units = 'cm')

## Plot: BP intervals
dat_RESULTS <- read_excel(path = 'ChristinaThreeLayerResult.xlsx', sheet = 1)

dat_BPI.OY.SME5.L3 <- data.frame(
  Energy = dat_RESULTS[, 1]
  , Belief = dat_RESULTS[, 5]
  , Plausibility = dat_RESULTS[, 4])

colnames(dat_BPI.OY.SME5.L3) <- c('Energy', 'Belief', 'Plausibility')

dat.long_BPI.OY.SME5.L3 <- tidyr::pivot_longer(
  data = dat_BPI.OY.SME5.L3
  , cols = c(Belief, Plausibility)
  , names_to = 'Bounds'
  , values_to = 'CDF')

plot_BPI.OY.SME5.L3 <- ggplot(
  dat.long_BPI.OY.SME5.L3, aes(x = Energy, y = CDF, colour = Bounds, group = Bounds)) +
  geom_step(size = 2) +
  scale_color_manual(values = c('orange', 'purple')) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Energy (ft lbf), x') +
  ylab('P(X <= x) as BPI') +
  scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
  geom_hline(yintercept = 0.1, linetype = 'dashed', color = 'red', size = 1.5) +
  geom_vline(xintercept = 453, linetype = 'dashed', color = 'blue', size = 1.5) +
  geom_vline(xintercept = 827, linetype = 'dashed', color = 'green', size = 1.5) +
  annotate(geom = 'label', label = 'P(X <= x) = 0.10', x = 700, y = 0.15) +
  annotate(geom = 'label', label = 'x = 453 ft lbf', x = 453, y = 0.40) +
  annotate(geom = 'label', label = 'x = 827 ft lbf', x = 827, y = 0.40) +
  theme(legend.position = c(0.05, 0.95), legend.justification = c(0, 1))

print(plot_BPI.OY.SME5.L3)

ggsave(filename = 'BPI.OY.SME5.L3.png', plot = plot_BPI.OY.SME5.L3, device = 'png', width
= 15, height = 15, units = 'cm')


###############################
#### Empirical CDF section ####
###############################

## Set-up functions
my.seed <- 34567
set.seed(my.seed)

f_Layer <- function(R = 1e4
    , distr = list(runif, runif)
    , param = list(c(500, 600), c(500, 600))
    , wts   = list(0.7, 0.3)
    ) {
    n <- R * unlist(wts)
    result <- list()
    for (i_distr in 1:length(distr)) {
      if (unlist(n[[i_distr]]) == 0) {
        result[[i_distr]] <- numeric(0)
        next
      }
```

```
# If statements are used depending on how many parameters the distribution needs
      if (length(param[[i_distr]]) == 1) {
        result[[i_distr]] <-
          distr[[i_distr]](
            n[[i_distr]]
            , param[[i_distr]][[1]]
          )
      }
      if (length(param[[i_distr]]) == 2) {
        result[[i_distr]] <-
          distr[[i_distr]](
            n[[i_distr]]
            , param[[i_distr]][[1]]
            , param[[i_distr]][[2]]
          )
      }
      if (length(param[[i_distr]]) == 3) {
        result[[i_distr]] <-
          distr[[i_distr]](
            n[[i_distr]]
            , param[[i_distr]][[1]]
            , param[[i_distr]][[2]]
            , param[[i_distr]][[3]]
          )
      }
      if (length(param[[i_distr]]) == 4) {
        result[[i_distr]] <-
          distr[[i_distr]](
            n[[i_distr]]
            , param[[i_distr]][[1]]
            , param[[i_distr]][[2]]
            , param[[i_distr]][[3]]
            , param[[i_distr]][[4]]
          )
      }
      if (length(param[[i_distr]]) == 5) {
        result[[i_distr]] <-
          distr[[i_distr]](
            n[[i_distr]]
            , param[[i_distr]][[1]]
            , param[[i_distr]][[2]]
            , param[[i_distr]][[3]]
            , param[[i_distr]][[4]]
            , param[[i_distr]][[5]]
          )
      }

    }

    result <- unlist(result)
    return(result)
  }

f_norm_param_from_bounds <-
  function(
    lower_bound = 00.00   # Lower bound
    , upper_bound = 08.00 # Upper bound
    , std         = 2     # Number of standard deviations to go out
  ) {

    out_param <-
      list(mu = mean(c(lower_bound, upper_bound))
        , sd      = (mean(c(lower_bound, upper_bound)) - lower_bound) / std
        , lb      = lower_bound
        , ub      = upper_bound
        , method  = "invtransfo"
      )
    return(out_param)
  }
```

```
set.seed(76543)
R    <- 1e4
eCDF <- list()

n_examples <- c("BPI_ON_SME1_L1", "BPI_OY_SME1_L1", "BPI_OY_SME1_L2")
n_dist     <- c("Uniform", "Normal", "Exponential")

for (i_examples in n_examples) {
  eCDF[[i_examples]] <- list()
  for (i_dist in n_dist) {
    eCDF[[i_examples]][[i_dist]] <- list()
  }
}

## EX: Overlapping = N, SME(s) = 1, Layer(s) = 1
dat_BPI_ON_SME1_L1 <-
  data.frame(Time = c(
        as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 10:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 12:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 15:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 18:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 23:59:59', format = '%Y-%m-%d %H:%M:%S')
      )
    , Belief = c(0, 0, 0.1, 0.2, 0.5, 0.9, 1, 1)
    , Plausibility = c(0, 0.1, 0.2, 0.5, 0.9, 1, 1, 1)
  )

dat.long_BPI_ON_SME1_L1 <-
  tidyr::pivot_longer(
    data = dat_BPI_ON_SME1_L1
    , cols = c(Belief, Plausibility)
    , names_to = 'Bounds'
    , values_to = 'CDF'
  )

eCDF[["BPI_ON_SME1_L1"]][["Uniform"]] <-
  f_Layer(
    R = R
    , distr = list(runif, runif, runif, runif, runif, runif, runif)
    , param = list(
      c(00.00, 08.00)
      , c(08.00, 10.00)
      , c(10.00, 12.00)
      , c(12.00, 15.00)
      , c(15.00, 18.00)
      , c(18.00, 20.00)
      , c(20.00, 24.00)
    )
    , wts   = list(0, 0.1, 0.1, 0.3, 0.4, 0.1, 0)
  )

eCDF[["BPI_ON_SME1_L1"]][["Normal"]] <-
  f_Layer(
    R = R
    , distr = list(rtnorm, rtnorm, rtnorm, rtnorm, rtnorm, rtnorm, rtnorm)
    , param = list(
      f_norm_param_from_bounds(00.00, 08.00)
      , f_norm_param_from_bounds(08.00, 10.00)
      , f_norm_param_from_bounds(10.00, 12.00)
      , f_norm_param_from_bounds(12.00, 15.00)
      , f_norm_param_from_bounds(15.00, 18.00)
      , f_norm_param_from_bounds(18.00, 20.00)
      , f_norm_param_from_bounds(20.00, 24.00)
    )
    , wts   = list(0, 0.1, 0.1, 0.3, 0.4, 0.1, 0)
  )

exp_rate  <- 1
```

```
exp_scale <- 10
exp_rate  <- exp_rate * exp_scale
eCDF[["BPI_ON_SME1_L1"]][["Exponential"]] <- f_Layer(
    R = R
    , distr = list(rexptr, rexptr, rexptr, rexptr, rexptr, rexptr, rexptr)
    , param = list(
      list(exp_rate, c(00.00, 08.00) / exp_scale)
      , list(exp_rate, c(08.00, 10.00) / exp_scale)
      , list(exp_rate, c(10.00, 12.00) / exp_scale)
      , list(exp_rate, c(12.00, 15.00) / exp_scale)
      , list(exp_rate, c(15.00, 18.00) / exp_scale)
      , list(exp_rate, c(18.00, 20.00) / exp_scale)
      , list(exp_rate, c(20.00, 24.00) / exp_scale)
    )
    , wts = list(0, 0.1, 0.1, 0.3, 0.4, 0.1, 0)
  ) * exp_scale

eCDF_long <- tidyr::pivot_longer(
    data = eCDF[["BPI_ON_SME1_L1"]] %>% as_tibble()
    , cols = everything()
    , names_to  = "Distributions"
    , values_to = "Sample"
  ) %>%
  arrange(
    Distributions
    , Sample
  ) %>%
  mutate(
    Distributions = Distributions %>% factor()
    , Time = as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S') + 3600 *
Sample
  ) %>%
  group_by(
    Distributions
  ) %>%
  mutate(
    eCDF = (1:n()) / n()
  ) %>%
  ungroup() %>%
  rename(
    Bounds = Distributions
    , CDF      = eCDF
  )

dat.long_BPI_ON_SME1_L1 <-
  dat.long_BPI_ON_SME1_L1 %>%
  bind_rows(
    eCDF_long
  ) %>%
  mutate(
    Bounds = Bounds %>% factor()
  )

plot_BPI_ON_SME1_L1 <-
  ggplot(dat.long_BPI_ON_SME1_L1 %>% filter(Bounds == c("Belief", "Plausibility"))
         , aes(x = Time, y = CDF, colour = Bounds, group = Bounds)) +
  geom_step(size = 2) +
  scale_x_datetime(breaks = '2 hours', date_labels = '%H:%M') +
  scale_color_manual(
    breaks=c("Belief", "Plausibility", "Uniform", "Normal", "Exponential")
    , values = c('orange', 'purple', "blue", "red", "green")
    , name = "Bounds or\nDistribution",
  ) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Delivery Time (Military)') +
  ylab('P(X <= x) as BPI') +
  scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
  theme(legend.position = c(0.05, 0.95), legend.justification = c(0, 1))

plot_BPI_ON_SME1_L1 <-
```

```
  plot_BPI_ON_SME1_L1 +
  geom_step(data = eCDF_long %>% filter(Bounds == c("Uniform", "Normal", "Exponential"))
            , size = 1, alpha = 1)

print(plot_BPI_ON_SME1_L1)

ggsave(filename = 'BPI_ON_SME1_L1.png', plot = plot_BPI_ON_SME1_L1, device = 'png', width
= 15, height = 15, units = 'cm')

## EX: Overlapping = Y, SME(s) = 1, Layer(s) = 1
dat_BPI_OY_SME1_L1 <-
  data.frame(
    Time =
      c(
        as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 08:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 09:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 10:30:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 11:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 13:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 14:30:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 15:30:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 17:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 18:30:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 20:00:00', format = '%Y-%m-%d %H:%M:%S')
        , as.POSIXct('2020-01-01 23:59:59', format = '%Y-%m-%d %H:%M:%S')
      )
    , Belief = c(0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.5, 0.5, 0.9, 1, 1)
    , Plausibility = c(0, 0.1, 0.2, 0.2, 0.5, 0.5, 0.9, 0.9, 1, 1, 1, 1)
  )

dat.long_BPI_OY_SME1_L1 <-
  tidyr::pivot_longer(
    data = dat_BPI_OY_SME1_L1
    , cols = c(Belief, Plausibility)
    , names_to = 'Bounds'
    , values_to = 'CDF'
  )

eCDF[["BPI_OY_SME1_L1"]][["Uniform"]] <-
  f_Layer(
    R = R
    , distr = list(runif, runif, runif, runif, runif, runif, runif)
    , param = list(
      c(00.00, 08.00)
      , c(08.00, 10.50)
      , c(09.00, 13.00)
      , c(11.00, 15.50)
      , c(14.50, 18.50)
      , c(17.00, 20.00)
      , c(20.00, 24.00)
    )
    , wts = list(0, 0.1, 0.1, 0.3, 0.4, 0.1, 0)
  )

eCDF[["BPI_OY_SME1_L1"]][["Normal"]] <-
  f_Layer(
    R = R
    , distr = list(rtnorm, rtnorm, rtnorm, rtnorm, rtnorm, rtnorm, rtnorm)
    , param = list(
      f_norm_param_from_bounds(00.00, 08.00)
      , f_norm_param_from_bounds(08.00, 10.50)
      , f_norm_param_from_bounds(09.00, 13.00)
      , f_norm_param_from_bounds(11.00, 15.50)
      , f_norm_param_from_bounds(14.50, 18.50)
      , f_norm_param_from_bounds(17.00, 20.00)
      , f_norm_param_from_bounds(20.00, 24.00)
    )
    , wts = list(0, 0.1, 0.1, 0.3, 0.4, 0.1, 0)
  )
```

```
exp_rate  <- 1
exp_scale <- 10
exp_rate  <- exp_rate * exp_scale
eCDF[["BPI_OY_SME1_L1"]][["Exponential"]] <-
  f_Layer(
    R = R
    , distr = list(rexptr, rexptr, rexptr, rexptr, rexptr, rexptr, rexptr)
    , param = list(
      list(exp_rate, c(00.00, 08.00) / exp_scale)
      , list(exp_rate, c(08.00, 10.50) / exp_scale)
      , list(exp_rate, c(09.00, 13.00) / exp_scale)
      , list(exp_rate, c(11.00, 15.50) / exp_scale)
      , list(exp_rate, c(14.50, 18.50) / exp_scale)
      , list(exp_rate, c(17.00, 20.00) / exp_scale)
      , list(exp_rate, c(20.00, 24.00) / exp_scale)
    )
    , wts   = list(0, 0.1, 0.1, 0.3, 0.4, 0.1, 0)
  ) * exp_scale

eCDF_long <-
  tidyr::pivot_longer(
    data = eCDF[["BPI_OY_SME1_L1"]] %>% as_tibble()
    , cols = everything()
    , names_to  = "Distributions"
    , values_to = "Sample"
  ) %>%
  arrange(
    Distributions
    , Sample
  ) %>%
  mutate(
    Distributions = Distributions %>% factor()
    , Time = as.POSIXct('2020-01-01 00:00:00', format = '%Y-%m-%d %H:%M:%S') + 3600 *
Sample
  ) %>%
  group_by(
    Distributions
  ) %>%
  mutate(
    eCDF = (1:n()) / n()
  ) %>%
  ungroup() %>%
  rename(
    Bounds = Distributions
    , CDF = eCDF
  )

dat.long_BPI_OY_SME1_L1 <-
  dat.long_BPI_OY_SME1_L1 %>%
  bind_rows(
    eCDF_long
  ) %>%
  mutate(
    Bounds = Bounds %>% factor()
  )

plot_BPI_OY_SME1_L1 <-
  ggplot(dat.long_BPI_OY_SME1_L1 %>% filter(Bounds == c("Belief", "Plausibility"))
         , aes(x = Time, y = CDF, colour = Bounds, group = Bounds)) +
  geom_step(size = 2) +
  scale_x_datetime(breaks = '2 hours', date_labels = '%H:%M') +
  scale_color_manual(
    breaks=c("Belief", "Plausibility", "Uniform", "Normal", "Exponential")
    , values = c('orange', 'purple', "blue", "red", "green")
    , name = "Bounds or\nDistribution",
  ) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Delivery Time (Military), x') +
  ylab('P(X <= x) as BPI') +
  scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
```

```
  theme(legend.position = c(0.05, 0.90), legend.justification = c(0, 1)) +
  geom_hline(yintercept = 0.95, linetype = 'dashed', color = 'red', size = 1.5) +
  annotate(geom = 'label', x = as.POSIXct('2020-01-01 10:00:00', format = '%Y-%m-%d
%H:%M:%S'), y = 1, label = 'P(X <= x) = 0.95')

plot_BPI_OY_SME1_L1 <-
  plot_BPI_OY_SME1_L1 +
  geom_step(data = eCDF_long %>% filter(Bounds == c("Uniform", "Normal", "Exponential"))
            , size = 1, alpha = 1)

print(plot_BPI_OY_SME1_L1)

ggsave(filename = 'BPI_OY_SME1_L1.png', plot = plot_BPI_OY_SME1_L1, device = 'png', width
= 15, height = 15, units = 'cm')

## EX: Overlapping = Y, SME(s) = 1, Layer(s) = 2
dat_BPI_OY_SME1_L2 <-
  data.frame(
    Energy = c(0, 820, 900, 980, 1300, 1400, 1500, 2000)
    , Belief = c(0, 0, 0, 0, 0.42, 0.88, 1, 1)
    , Plausibility = c(0, 0.12, 0.58, 1, 1, 1, 1, 1)
  )

dat.long_BPI_OY_SME1_L2 <-
  tidyr::pivot_longer(
    data = dat_BPI_OY_SME1_L2
    , cols = c(Belief, Plausibility)
    , names_to = 'Bounds'
    , values_to = 'CDF'
  )

eCDF[["BPI_OY_SME1_L2"]][["Uniform"]] <-
  f_Layer(
    R = R
    , distr = list(runif, runif, runif, runif, runif, runif)
    , param = list(
      c(0, 8.20)
      , c(9.80, 13.00)
      , c(8.20, 15.00)
      , c(9.00, 14.00)
      , c(9.00, 14.00)
      , c(15.00, 20.00)
    )
    , wts = list(0, 0.42, 0.12, 0.28, 0.18, 0)
  )

eCDF[["BPI_OY_SME1_L2"]][["Normal"]] <-
  f_Layer(
    R = R
    , distr = list(rtnorm, rtnorm, rtnorm, rtnorm, rtnorm, rtnorm)
    , param = list(
      f_norm_param_from_bounds(0, 8.20)
      , f_norm_param_from_bounds(9.80, 13.00)
      , f_norm_param_from_bounds(8.20, 15.00)
      , f_norm_param_from_bounds(9.00, 14.00)
      , f_norm_param_from_bounds(9.00, 14.00)
      , f_norm_param_from_bounds(15.00, 20.00)
    )
    , wts = list(0, 0.42, 0.12, 0.28, 0.18, 0)
  )

exp_rate  <- 1
exp_scale <- 10
exp_rate  <- exp_rate * exp_scale
eCDF[["BPI_OY_SME1_L2"]][["Exponential"]] <-
  f_Layer(
    R = R
    , distr = list(rexptr, rexptr, rexptr, rexptr, rexptr, rexptr)
    , param = list(
      list(exp_rate, c(0, 8.20) / exp_scale)
      , list(exp_rate, c(9.80, 13.00) / exp_scale)
```

```
      , list(exp_rate, c(8.20, 15.00) / exp_scale)
      , list(exp_rate, c(9.00, 14.00) / exp_scale)
      , list(exp_rate, c(9.00, 14.00) / exp_scale)
      , list(exp_rate, c(15.00, 20.00) / exp_scale)
    )
    , wts = list(0, 0.42, 0.12, 0.28, 0.18, 0)
  ) * exp_scale

eCDF_long <-
  tidyr::pivot_longer(
    data = eCDF[["BPI_OY_SME1_L2"]] %>% as_tibble()
    , cols = everything()
    , names_to  = "Distributions"
    , values_to = "Sample"
  ) %>%
  arrange(
    Distributions
    , Sample
  ) %>%
  mutate(
    Distributions = Distributions %>% factor()
    , Energy = Sample * 100
  ) %>%
  group_by(
    Distributions
  ) %>%
  mutate(
    eCDF = (1:n()) / n()
  ) %>%
  ungroup() %>%
  rename(
    Bounds = Distributions
    , CDF = eCDF
  )

dat.long_BPI_OY_SME1_L2 <-
  dat.long_BPI_OY_SME1_L2 %>%
  bind_rows(
    eCDF_long
  ) %>%
  mutate(
    Bounds = Bounds %>% factor()
  )

plot_BPI_OY_SME1_L2 <-
  ggplot(dat.long_BPI_OY_SME1_L2 %>% filter(Bounds == c("Belief", "Plausibility"))
         , aes(x = Energy, y = CDF, colour = Bounds, group = Bounds)) +
  geom_step(size = 2) +
  scale_color_manual(
    breaks=c("Belief", "Plausibility", "Uniform", "Normal", "Exponential")
    , values = c('orange', 'purple', "blue", "red", "green")
    , name = "Bounds or\nDistribution",
  ) +
  theme_bw() +
  fill_palette('jco') +
  xlab('Energy (ft lbf), x') +
  ylab('P(X <= x) as BPI') +
  scale_y_continuous(breaks = seq(0, 1, by = 0.2)) +
  theme(legend.position = c(0.05, 0.95), legend.justification = c(0, 1))

plot_BPI_OY_SME1_L2 <-
  plot_BPI_OY_SME1_L2 +
  geom_step(data = eCDF_long %>% filter(Bounds == c("Uniform", "Normal", "Exponential"))
            , size = 1, alpha = 1)

print(plot_BPI_OY_SME1_L2)

ggsave(filename = 'BPI_OY_SME1_L2.png', plot = plot_BPI_OY_SME1_L2, device = 'png', width
= 15, height = 15, units = 'cm')
```