# Analysis under uncertainty

Sören Kemmann

© Fraunhofer IESE 2010

# Motivation

- Early application of safety analysis is crucial for high-integrity systems.
- Exact failure probabilities are difficult to justify for
  - software and
  - during design.
- We need a way to handle uncertainty in FTA.
- → Deductive quantitative FTA with probability intervals.

© Fraunhofer IESE 2010

# Critical points of embedded-systems FTA

Process measures alone cannot ensure system safety.

We need ways to assess system quality early in the design process

... to support design decisions and tradeoff analyses.

State of the practice is application of FTA after the fact,

... for certification purposes only.

Little evidence for single-point probabilities in FTA

... in early phases and for software.

Lots can be criticised in quantitative FTA, especially for software,

... so why don’t we make do with qualitative methods?

© Fraunhofer IESE 2010

# Why qualitative analysis falls (sometimes) short

Qualitative analysis does not support gradual predictions.

Qualitative analysis does not allow comparison of likelihoods.

Qualitative analysis does not facilitate prioritisation.

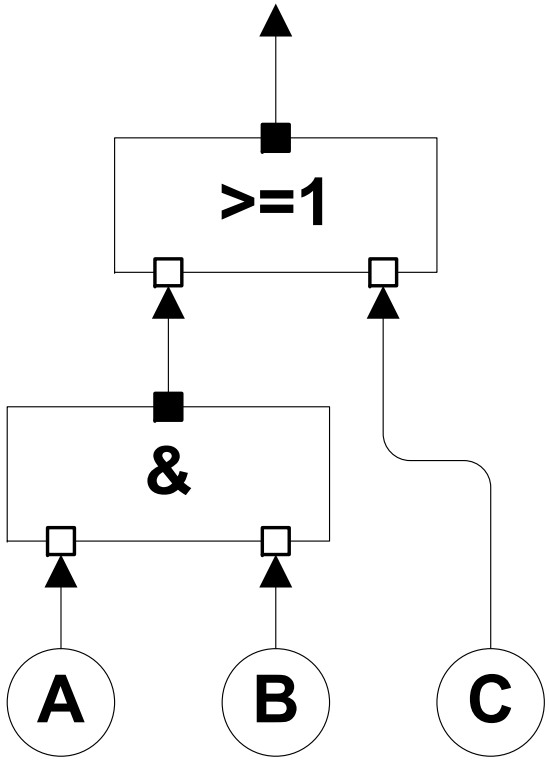
→ We would like to bridge the gap between qualitative and quantitative analysis!

© Fraunhofer IESE 2010

# Inductive vs deductive analysis

... to resulting probability.

From probability of causes




**From safety goal**

**... to probability of causes.**

The classical way – inductive quantitative FTA.

Our way – deductive quantitative FTA.

© Fraunhofer IESE 2010

# Three questions

(1) How probable is the fulfilment of the safety goal?

(If we do not have exact numbers.)

(2) How can we allocate basic-event reliabilities, such that the safety goal is met?

(3) What is the most efficient way to meeting the safety goal?

© Fraunhofer IESE 2010

# Handling uncertainty in FTA

The idea is simple, people do it all the time (well, safety analysts do).

What, if this basic event could not occur?

What, if that basic event were ten times as likely?

What, if we cut off a fault tree branch altogether?

→ What, if we play the “what-if” game a million times?

→ Probability intervals.

© Fraunhofer IESE 2010

# FTA with probability intervals

- Second-order probability

© Fraunhofer IESE 2010

# FTA with probability intervals

Numerical interval algebra does not work for us because we use symbols.

$X = [x_l, x_u]$, $Y = [y_l, y_u]$, $\bullet \in \{+, -, \times, \div\} \Rightarrow$
$X \bullet Y = [\min(x_l \bullet y_l, x_l \bullet y_u, x_u \bullet y_l, x_u \bullet y_u), \max(x_l \bullet y_l, x_l \bullet y_u, x_u \bullet y_l, x_u \bullet y_u)]$, $0 \in Y \Rightarrow \bullet \neq \div$ .

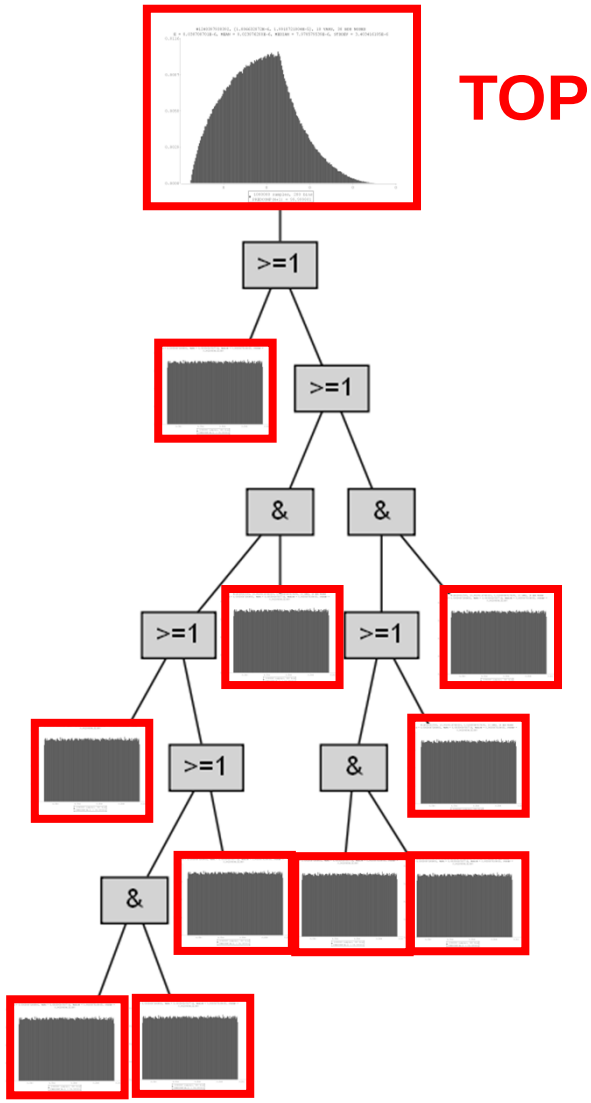
Excess may even be outside of [0, 1] → subdivision.

To avoid stochastic artifacts, we use random single-point sampling instead.

Answering question (1) → Getting to know fault tree personalities.

© Fraunhofer IESE 2010

# FTA with probability intervals




Leads to second-order probability mass functions.

→ Probability for the fault tree
having a certain probability when
basic events are assigned any probability
inside their bounds.

→ We use uniform distribution,
but any distribution is possible.

You can do it even with [0, 1] intervals!

→ characteristic PMF

© Fraunhofer IESE 2010

# FTA with probability intervals

Parameterised interval representation.

An interval $X = [x_l, x_u]$ is given by $f_{\alpha_x} = x_l + \alpha_x(x_u - x_l), \alpha_x \in [0, 1]$.

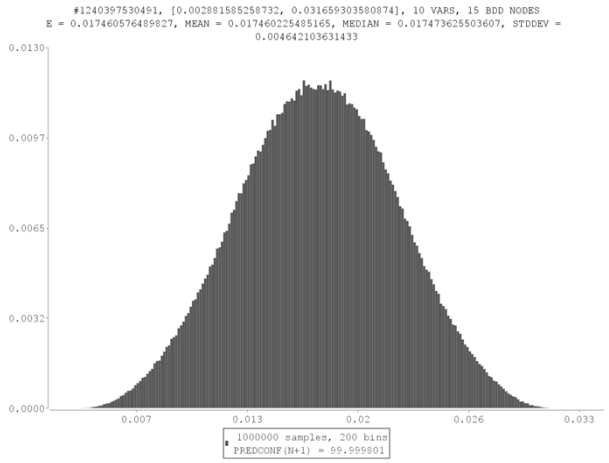
Calculating result bounds for noncoherent fault trees is NP hard!

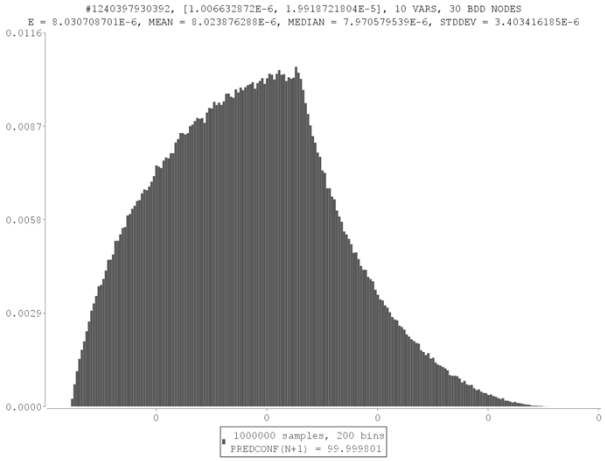
Expectation.

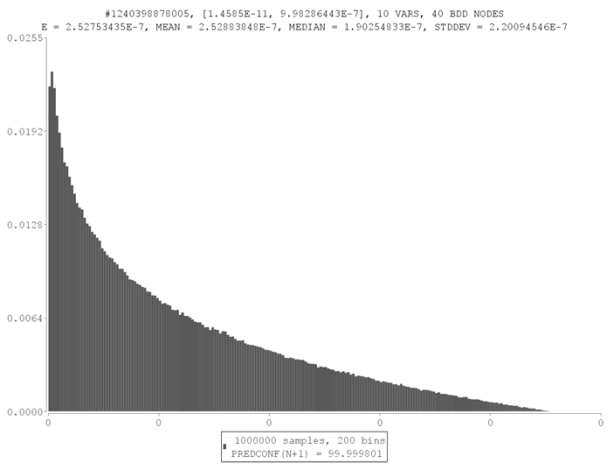
$$f = \sum \prod_{i \in I} X_i \Rightarrow E(f) = \sum \prod_{i \in I} X_i$$

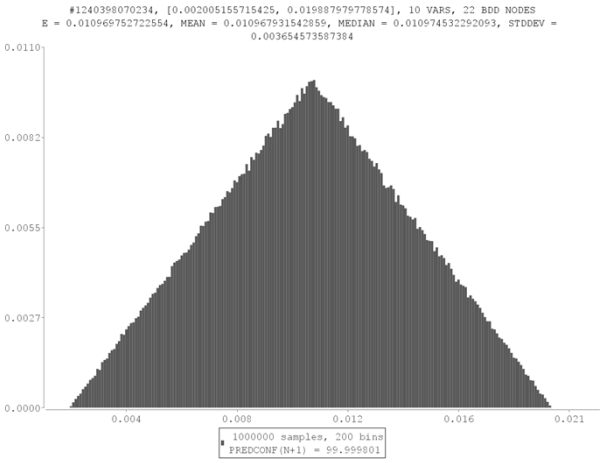
© Fraunhofer IESE 2010

# PMF of coherent fault trees – basic shapes



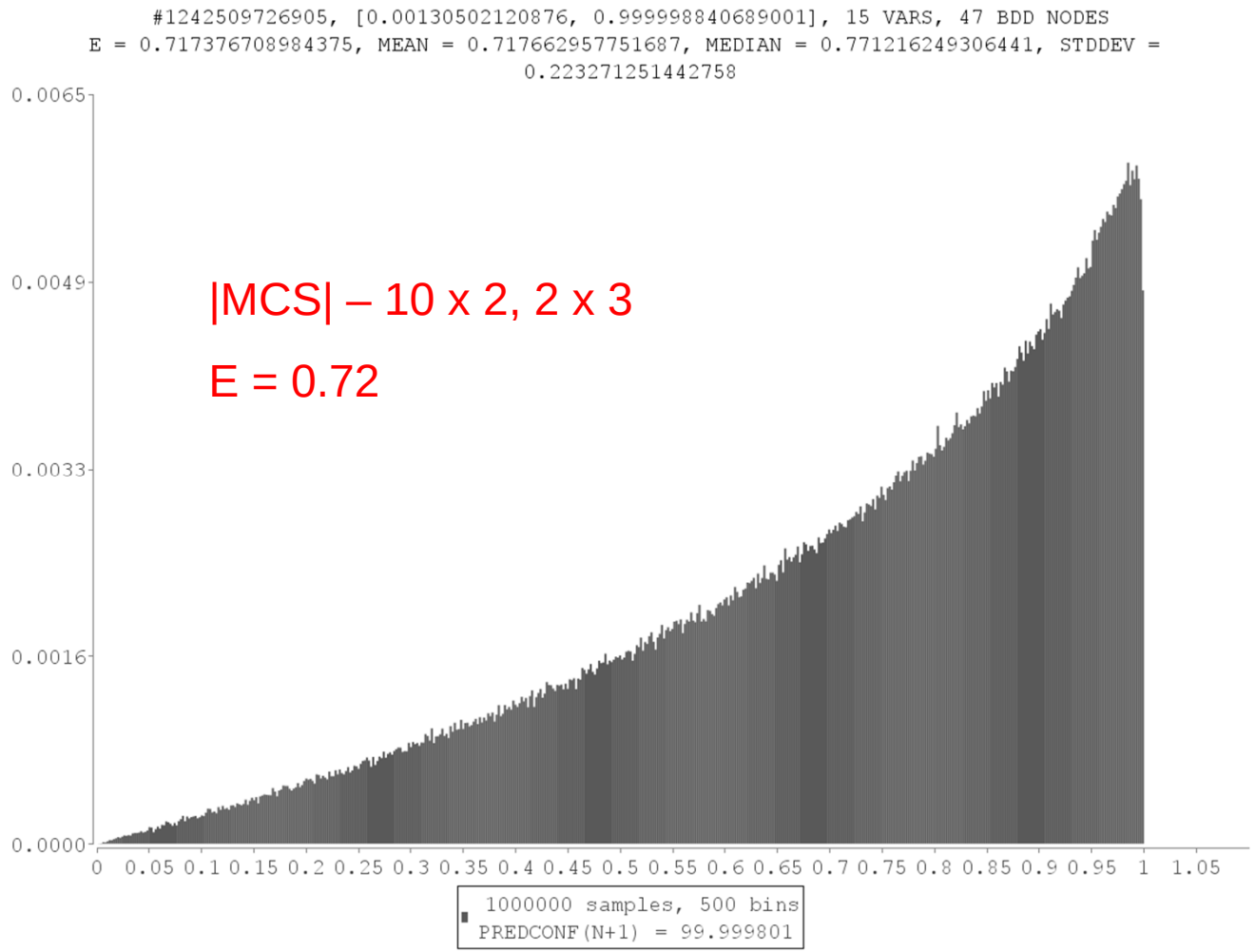




© Fraunhofer IESE 2010

Fraunhofer IESE

# Characteristic PMF



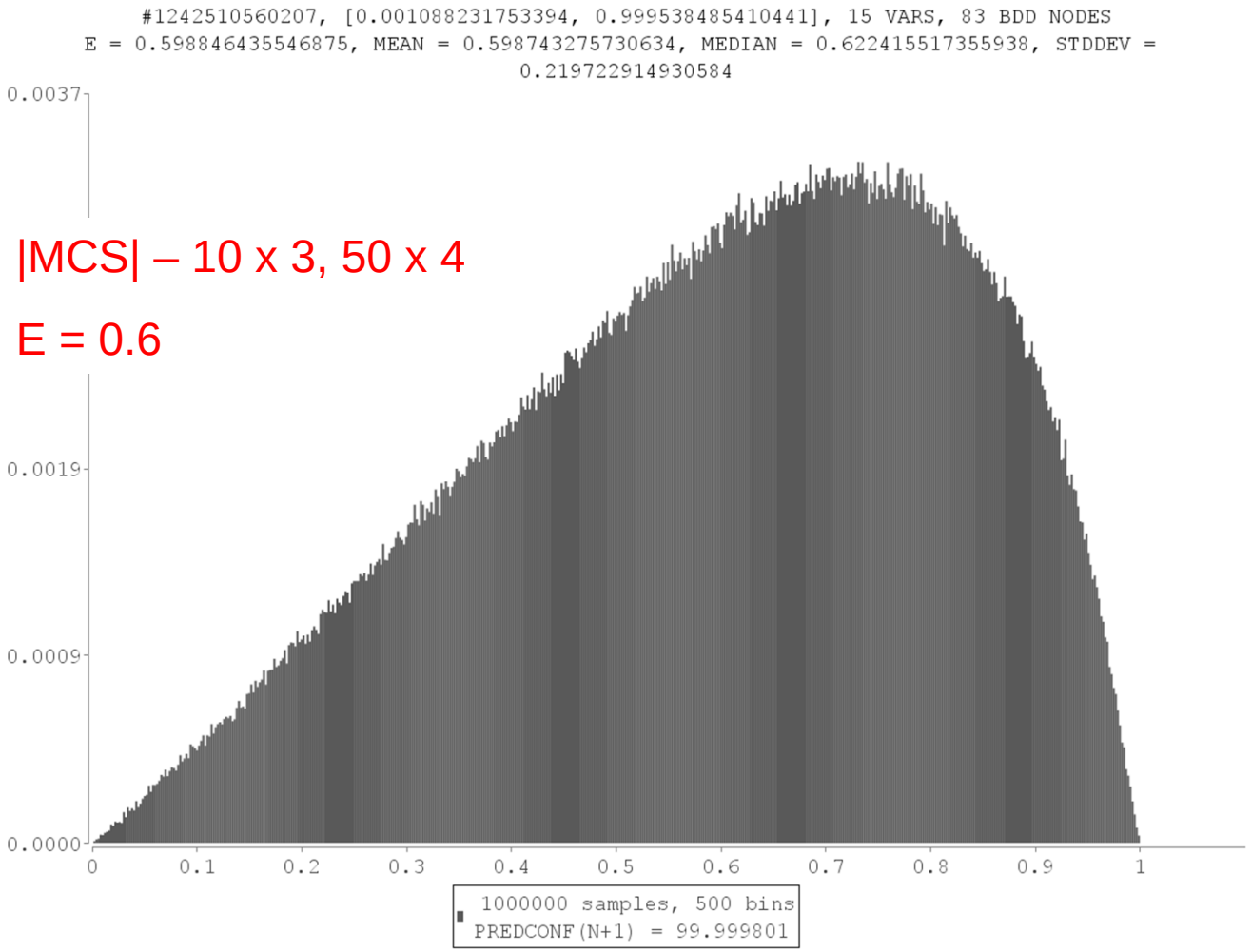

|MCS| – 10 x 2, 2 x 3

E = 0.72

© Fraunhofer IESE 2010

Fraunhofer IESE

# Characteristic PMF



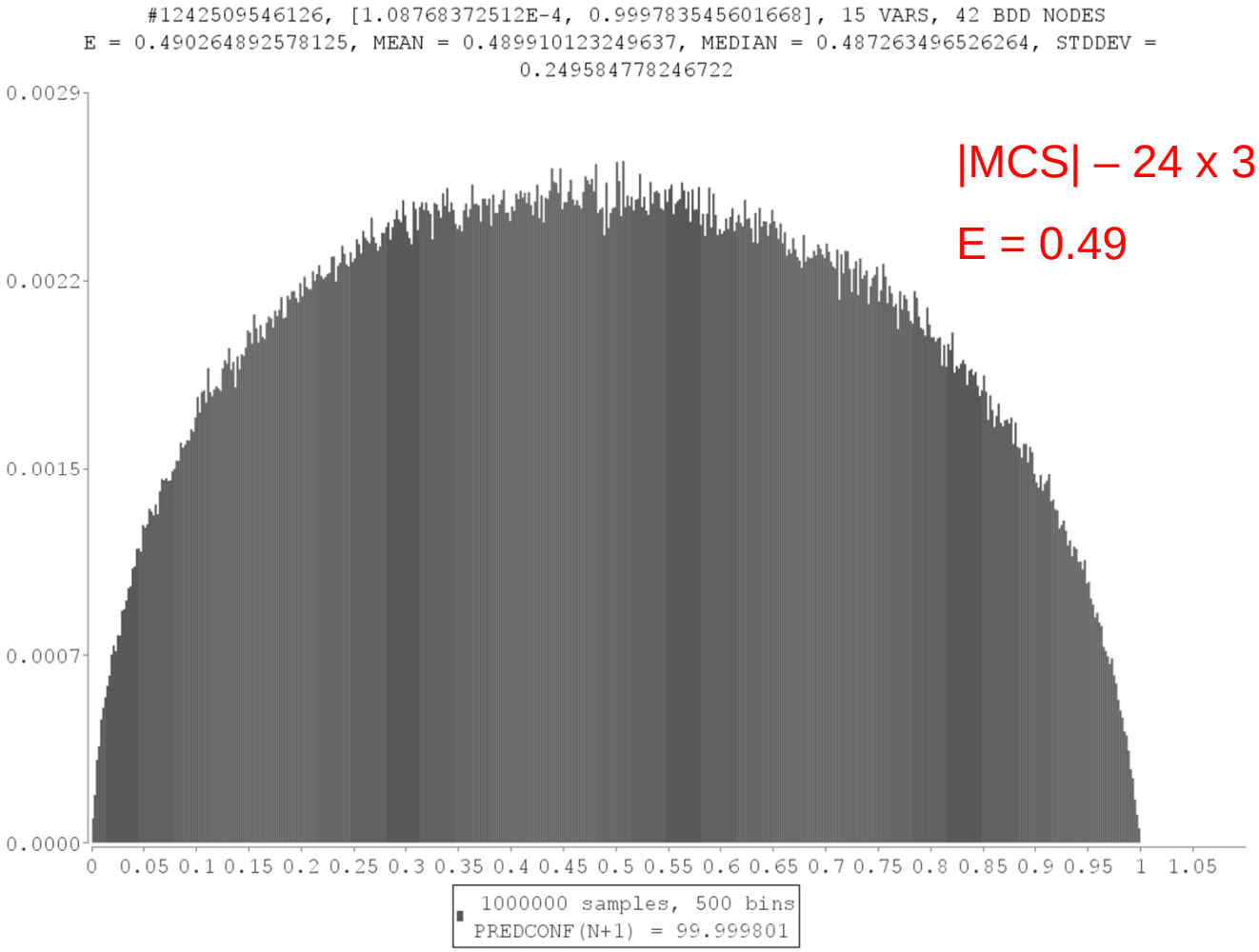

|MCS| – 10 x 3, 50 x 4

E = 0.6

© Fraunhofer IESE 2010

Fraunhofer IESE

# Characteristic PMF



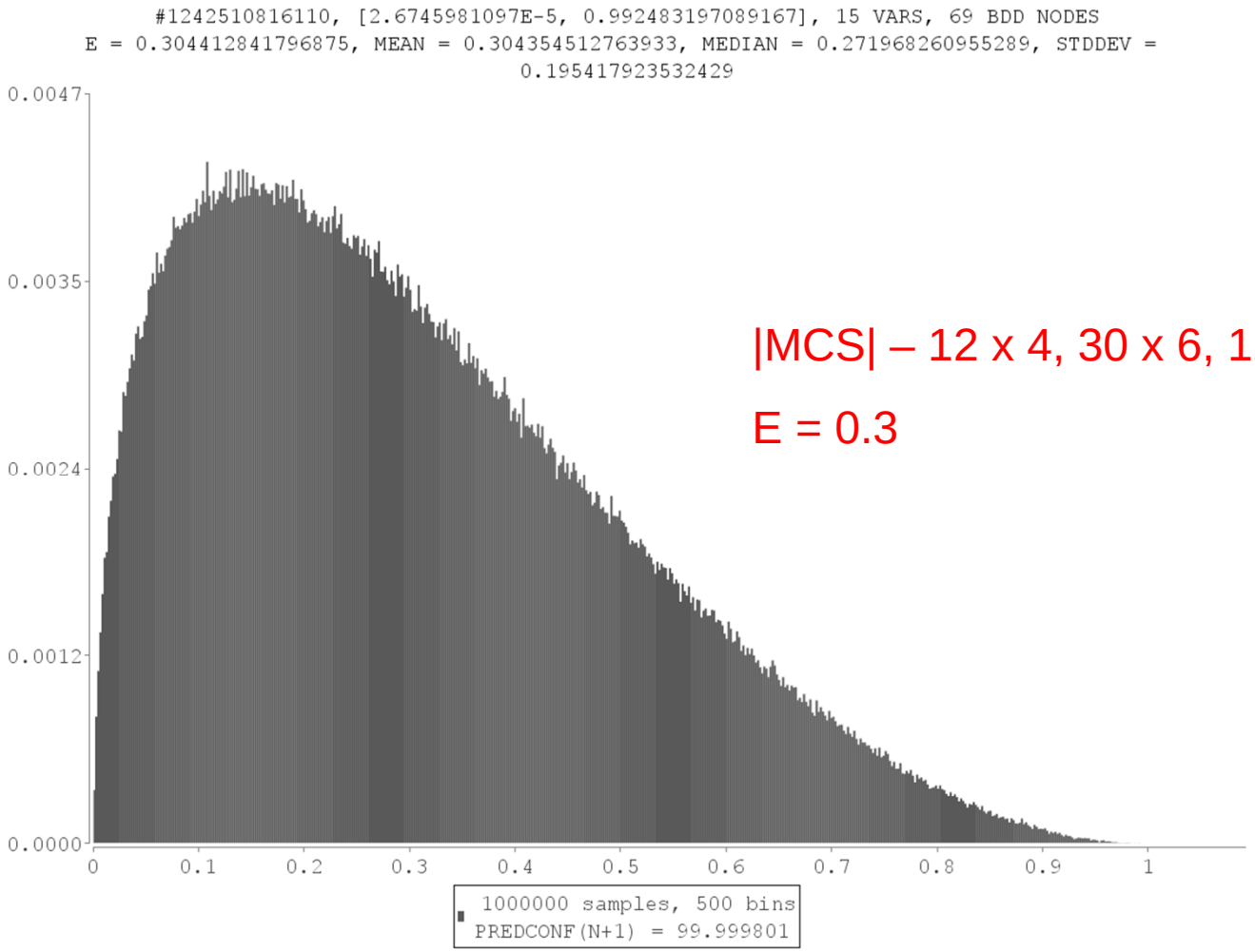

|MCS| – 24 x 3, 24 x 5

E = 0.49

© Fraunhofer IESE 2010

Fraunhofer IESE

# Characteristic PMF



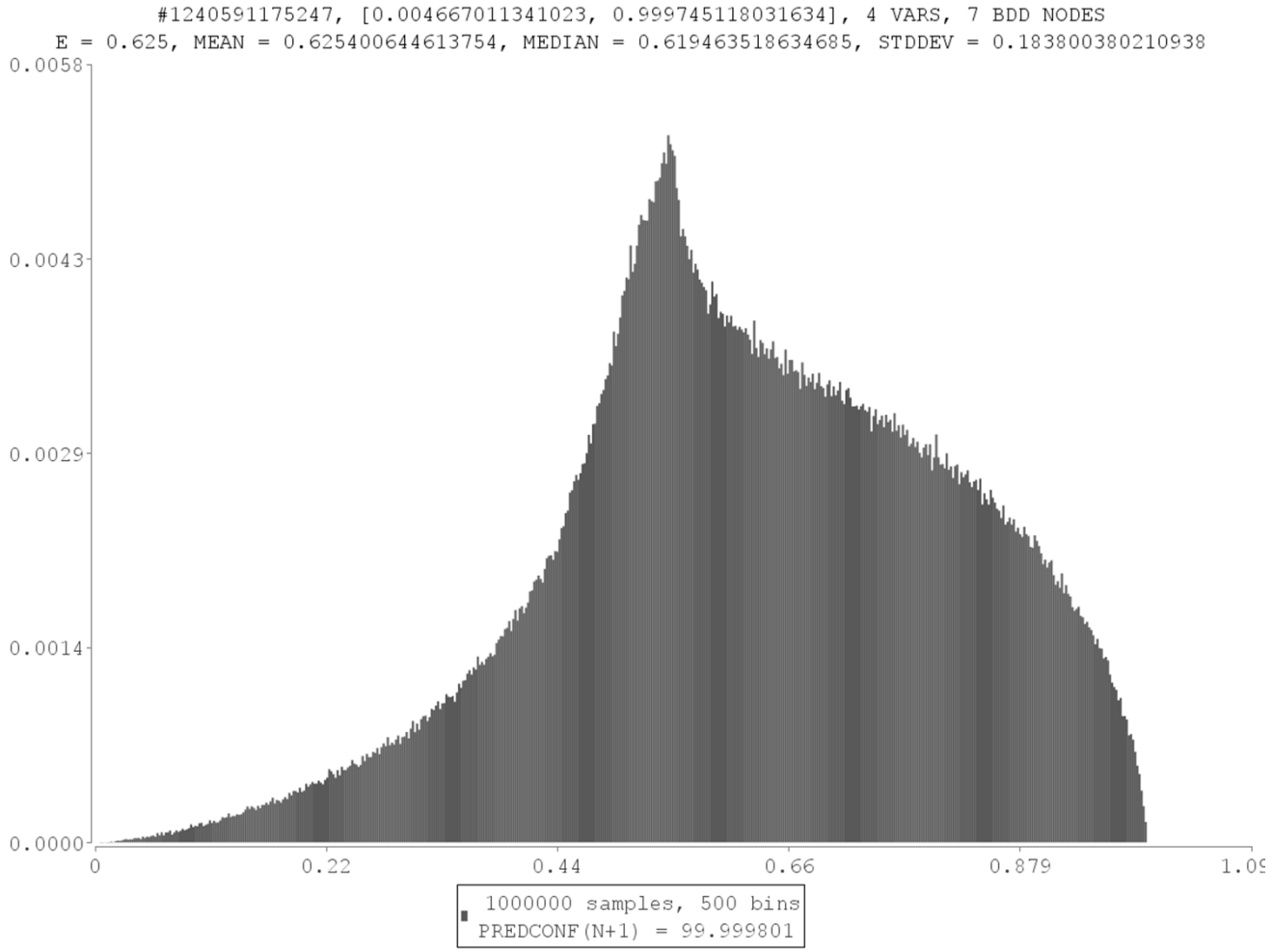

|MCS| – 12 x 4, 30 x 6, 12 x 8

E = 0.3

© Fraunhofer IESE 2010

Fraunhofer IESE

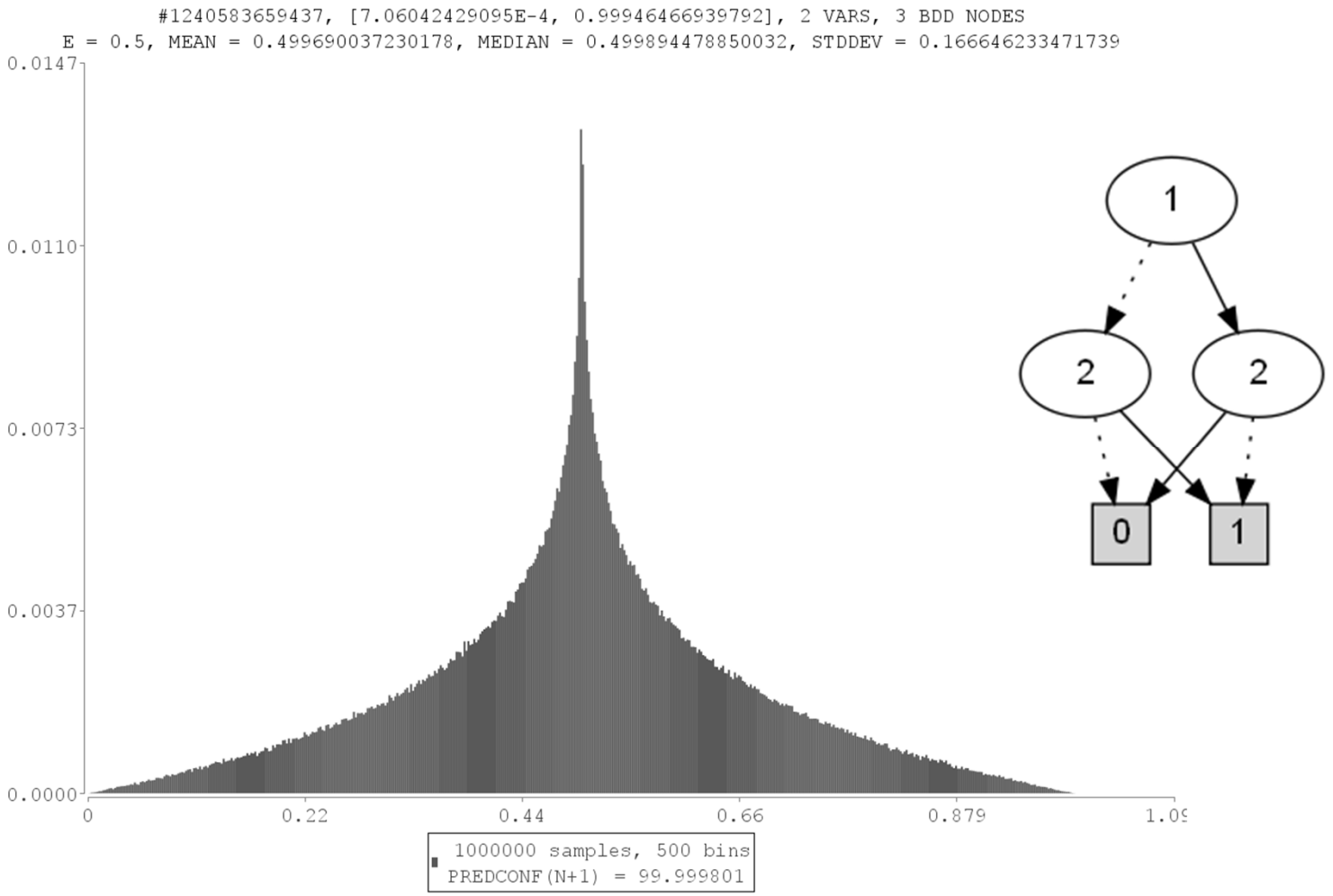
# PMF of noncoherent fault tree




© Fraunhofer IESE 2010

Fraunhofer IESE

# PMF of XOR




© Fraunhofer IESE 2010

Fraunhofer IESE

# The basic approach

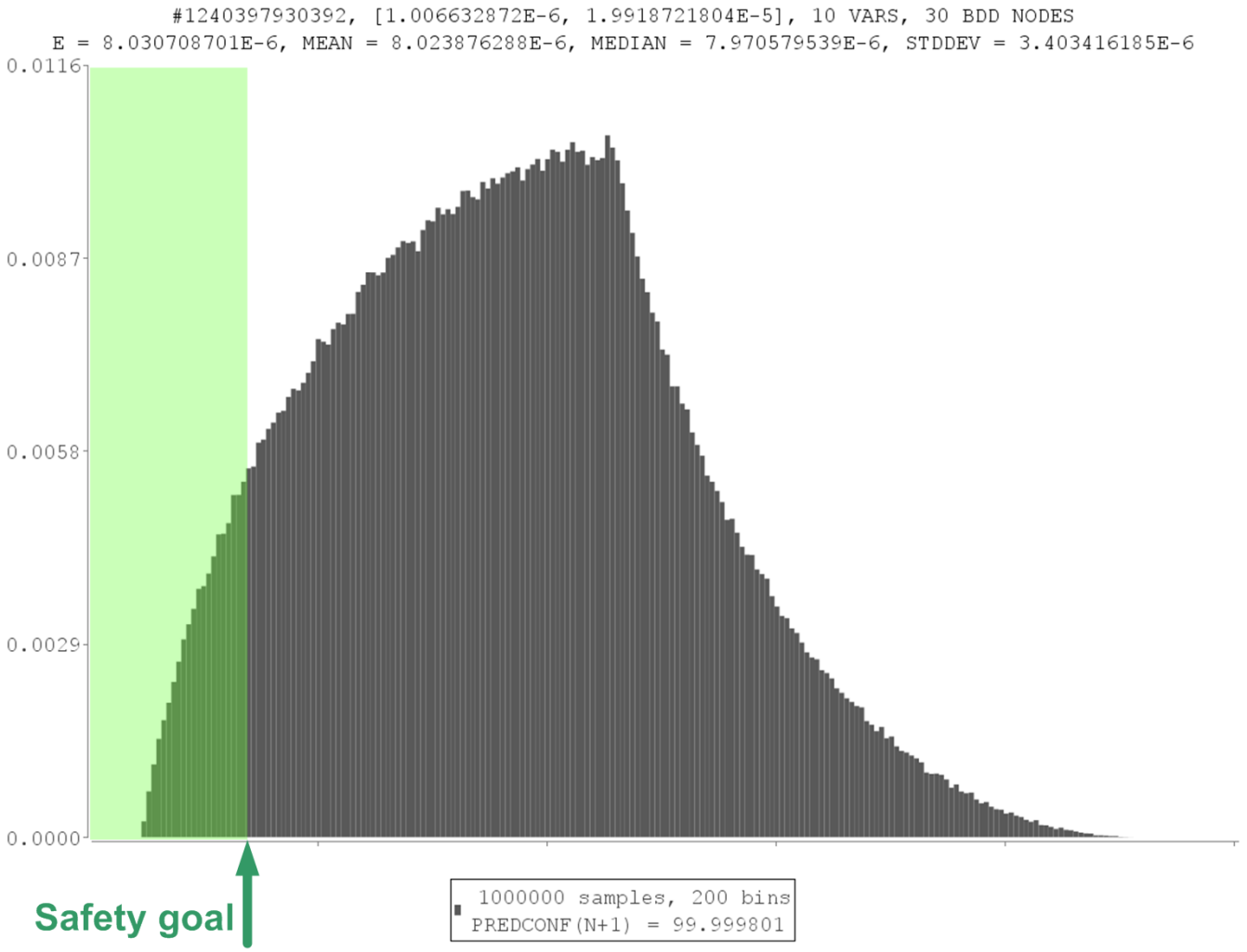
Answering question (2) → how can the safety goal be fulfilled?

Save alpha values for every sample and filter samples meeting the goal.

| **Sample no. / var.** | **1** | **2** | **3** | **4** | **5** | **6** | **...** |
|---|---|---|---|---|---|---|---|
| **1** | 0.36 | 0.44 | 0.1 | 0.77 | 0.29 | 0.56 | ... |
| **2** | 0.02 | 0.32 | 0.93 | 0.21 | 0.14 | 0.09 | ... |
| **...** | ... | ... | ... | ... | ... | ... | ... |

© Fraunhofer IESE 2010

# The basic approach




© Fraunhofer IESE 2010

Fraunhofer IESE

# Analysis with constraints

Answer question (3) → what is the most efficient way of meeting the safety goal?

Starting from the set of those samples that meet the safety goal

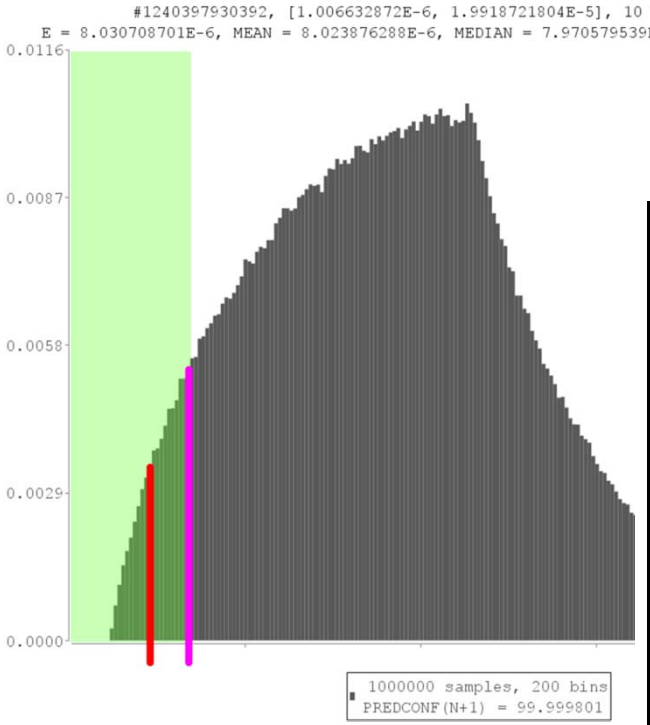
... apply a cost function to basic events,

... sort.

→ Cheapest sample still fulfilling the safety goal.

© Fraunhofer IESE 2010

# Analysis with constraints




MCS(f) = {{7}, {1, 5}, {2, 9}, {6, 9}, {3, 5, 8}, {4, 9, 10}}

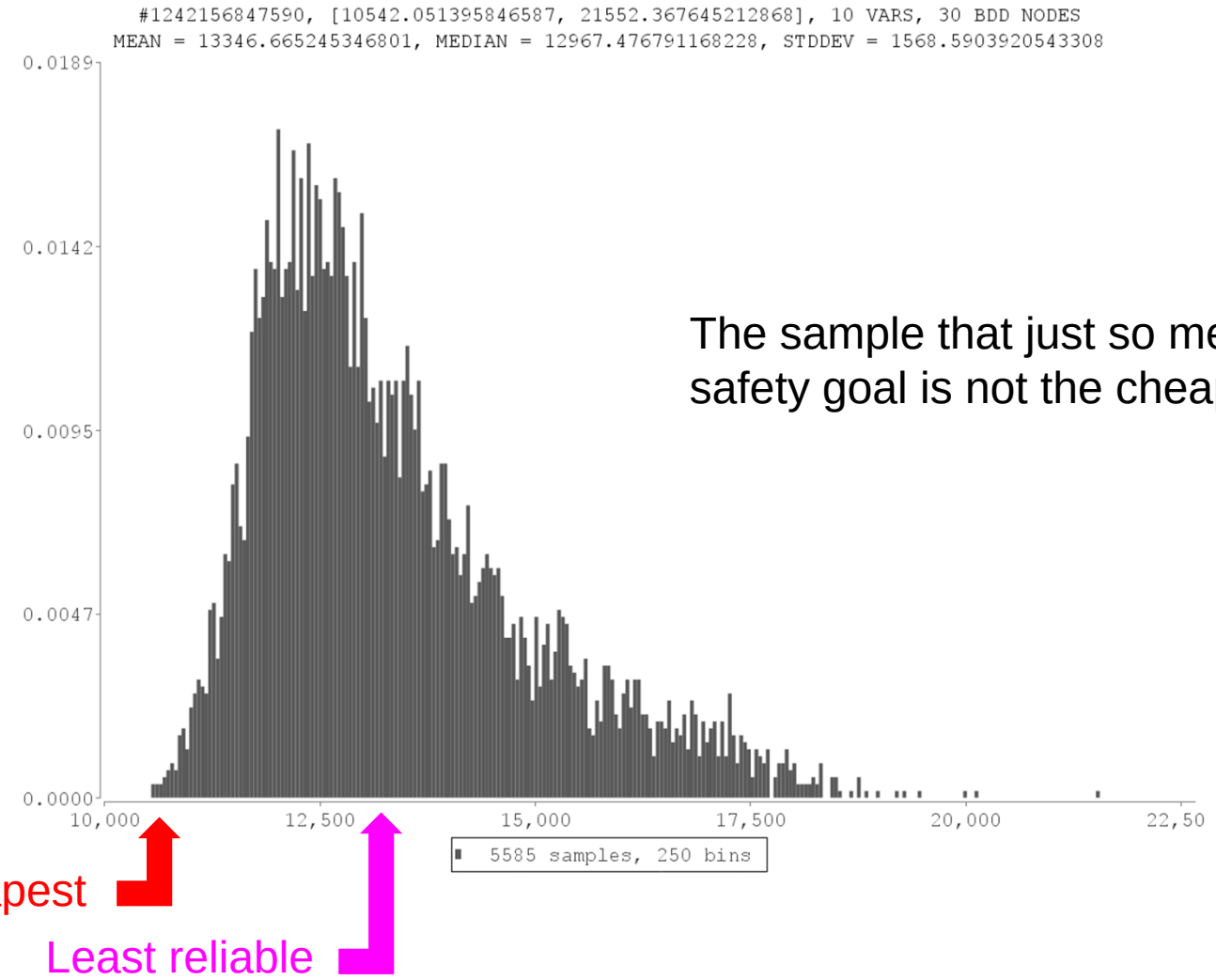
| *Var.* | *Pr. interval* | *∅ α* | *Sample 1, α* | *Sample 2, α* |
|---|---|---|---|---|
| **1** | **[$10^{-5}$, $10^{-2}$]** | **0.1928** | 0.0065 | 0.0012 |
| 2 | [$10^{-6}$, $10^{-4}$] | 0.4946 | 0.7283 | 0.5599 |
| 3 | [$10^{-5}$, $10^{-4}$] | 0.5028 | 0.2057 | 0.9086 |
| 4 | [$10^{-3}$, $10^{-2}$] | 0.5083 | 0.335 | 0.4909 |
| **5** | **[$10^{-5}$, $10^{-3}$]** | **0.229** | 0.0198 | 0.7988 |
| 6 | [$10^{-6}$, $10^{-3}$] | 0.4989 | 0.2062 | 0.3855 |
| **7** | **[$10^{-6}$, $10^{-5}$]** | **0.0192** | 0.0552 | 0.0488 |
| 8 | [$10^{-3}$, $10^{-2}$] | 0.4962 | 0.6478 | 0.2113 |
| 9 | [$10^{-6}$, $10^{-5}$] | 0.4977 | 0.2499 | 0.9643 |
| 10 | [$10^{-5}$, $10^{-3}$] | 0.5021 | 0.8567 | 0.887 |

## Cost function

$$f_C(sample_i) = \sum_{v_j \in V} \frac{j}{\sqrt{value(v_j(i))}}.$$

© Fraunhofer IESE 2010

# Analysis with constraints




The sample that just so meets the safety goal is not the cheapest.

© Fraunhofer IESE 2010

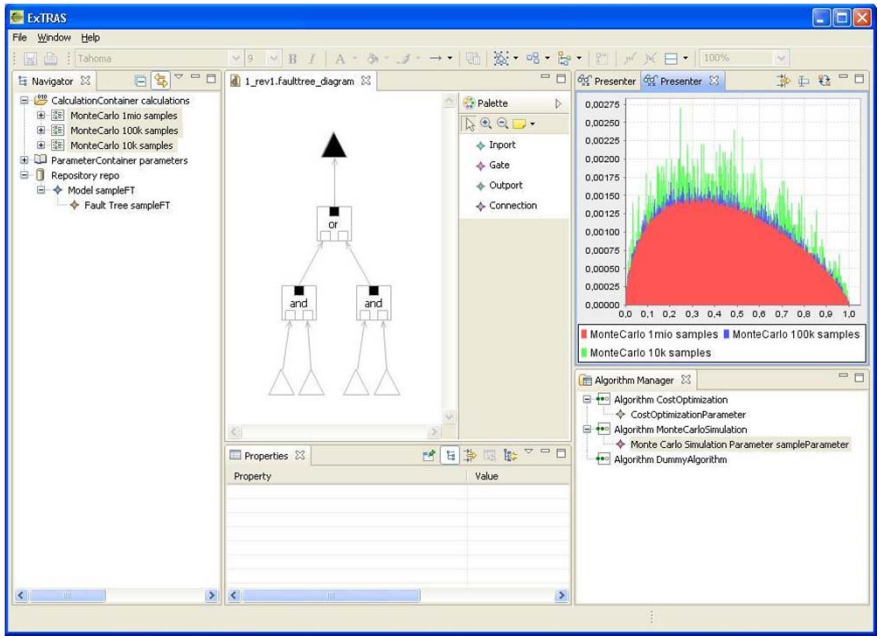
Fraunhofer IESE

# Conclusion

Second-order PMF allow to bridge gap between quantitative and qualitative FTA.

Facilitate integration of HW and SW FTA.

Guidance for design decisions and effort optimisation.




Intuitive visualisation of dependability/safety.

→ Common-cause analysis is important because of independence assumption.

→ Integration with component-based fault tree concept.

© Fraunhofer IESE 2010

Fraunhofer IESE

# Questions

- ...?

© Fraunhofer IESE 2010