What Is Optimal Granularity When Estimating Reliability of a Complex Engineering Systems

Niklas Winnewisser[1], Michael Beer[1],
Olga Kosheleva[2] and Vladik Kreinovich[2]

[1]Leibniz University Hannover, Hannover, Germany,
{winnewisser,beer}@irz.uni-hannover.de
[2]University of Texas at El Paso, El Paso, Texas 79968, USA
{olgak, vladik}@utep.edu

1. Practical problem

- In many real-life situations, we have a complex engineering system that consists of a large number of components.
- This may be a road network, this may be an electricity network, this may be a complex machine.
- We would like to know the reliability of this system, i.e.:
  - for every moment of time t,
  - we would like to know its survival probability $S(t)$ – the probability that the system will continued to perform its function up to moment t from now.
- The reliability of the system depends on how reliable are its components.
- So, to find the desired value $S(t)$, we need to know the survival probabilities $S_i(t)$ of its components.

2. Practical problem (cont-d)

- In many cases, these survival probabilities are described by the exponential distribution $S_i(t) = \exp(-\lambda_i \cdot t)$ for some value $\lambda_i > 0$.
- Based on the known values λ_i for all the components, we need to estimate the desired function $S(t)$.

3. This problem is usually estimated by simulations

- For simple systems, we sometimes have analytical expressions that describe the desired f-n $S(t)$ in terms of the known parameters λ_i.
- However, for complex system, such an expression is usually not possible.
- So, the only way to estimate $S(t)$ is by using simulations.
- Specifically:
  - for each moment t and for each i, we simulate whether this component works or not, with probability $\exp(-\lambda_i \cdot t)$,
  - and then we run the full simulation of the system to check whether this system will function or not.
- Once we repeat this simulation several times:
  - we can estimate the desired probability $S(t)$
  - by counting the proportion of cases in which the simulated system functioned.

4. This problem is usually estimated by simulations (cont-d)

- In general, if we run N simulations, we get the desired probability $S(t)$ with accuracy $\sim 1/\sqrt{N}$.
- So, to get 10% accuracy in estimating $S(t)$, we need 100 simulations; to get 20% accuracy, we need 25 simulations, etc.

5. Limitations of the simulations approach

- A complex engineering system contains a large number of components.
- So each simulation of this system requires a lot of computation time.
- And we need to repeat this simulation several times – 25 or 100.
- We also need to repeat these simulations for different moments of time t.
- The resulting time becomes very large – up to being not realistic.
- If the results show that the estimated probability of failure does not exceed the required threshold, we are good.
- However, often, it turns out that with the originally planned components, the estimated probability of failure is too high.
- So we need to replace some of the components by more reliable ones.
- To analyze the effect of this replacement on the system's reliability, we need to run the simulations again.

6. Limitations of the simulations approach (cont-d)

- And we may need to repeat this process again and again until we figure out how to satisfy the given reliability requirements.
- This need to repeat simulations again and again further increases the computation time, often making it astronomical and unrealistic.

7. What we can do: case when many components are identical to each other

- In many practical situations, the system:
  - consists of components of several types, and
  - all components of the same type have the same survival probability.
- In this case, we can speed up simulations as follows.
- Let s be the overall number of types, and let $n_1, \ldots, n_s$ be the number of components of each type.
- Then, for each combination of values $k_1 \le n_1, \ldots, k_s \le n_s$, we can use simulations to find the probability $P(k_1, \ldots, k_s)$ that the system will continue functioning if:
  - randomly selecting k_1 of n_1 components of Type 1 continue functioning, ..., and
  - randomly selected k_s components of type s continue functioning.

8. Case when many components are identical to each other (cont-d)

- Then, we can compute the probability that the system will survive by computing the value

$$S(t)=\sum_{k_1=1}^{n_1}\ldots\sum_{k_s=1}^{n_s}P(k_1,\ldots,k_s)\cdot P_1(k_1)\cdot\ldots\cdot P_s(k_s).\text{ where}$$

$$P_j(k_j)\stackrel{\text{def}}{=}\binom{n_j}{k_j}\cdot(S_j(t))^{k_j}\cdot(1-S_j(t))^{n_j-k_j}.$$

- The advantage of this approach is that:
  - if we need to replace one of the components with a more reliable one,
  - we do not need to repeat time-consuming simulations,
  - we can use the above formula to explicitly compute $S(t)$.

9. What can we do in the general case

- What if the full simulations would require too much time, but the components are not identical?
- A natural idea is:
  - to approximate the given situation
  - with a situation in which components with close characteristics are assumed to be identical.
- In other words:
  - we group (cluster) components into several clusters (granules), and
  - within each of these clusters j, we replace all different values λ_i with the common (in some sense, average) value $\widetilde{\lambda}_j$.

10. Resulting question

- Of course, the above scheme is approximate, since:
  - in our simulations,
  - we replace the original exact values λ_i with the approximate values $\widetilde{\lambda}_j$.
- It is desirable to come up with a clustering algorithm for which the inaccuracy caused by this replacement is the smallest possible.
- In this talk, we describe the optimal clustering algorithm for this problem.
- Here, we restrict ourselves to situations in which the reliability of each component is described by the exponential distribution.
- *Future work:* it is desirable to extend our clustering method:
  - to more general situations, in which
  - the reliability of some components is characterized by other distributions – e.g., by the Weibull law.

11. How inaccuracy of knowing $S_i(t)$ affects the inaccuracy of estimating $S(t)$

- To select the granulation that leads to the most accurate estimation for $S(t)$, we need to know:
  - how, in general, the inaccuracy of knowing each value $S_i(t)$
  - affects the inaccuracy of estimating $S(t)$.
- In general, the formula of full probability implies that

$$S(t) = S_i(t) \cdot P_i^+(t) + (1 - S_i(t)) \cdot P^-(t).$$

- Here, $P_i^+(t)$ is the probability that the system will function at moment t if the i-th component is still functional.
- And $P_i^-(t)$ is probability that the system will function at moment t if the i-th component has already failed.
- This expression can be rewritten as $S(t) = S_i(t) \cdot (P_i^+(t) - P_i^-(t)) + P_i^-(t)$.

12. How inaccuracy of knowing $S_i(t)$ affects the inaccuracy of estimating $S(t)$ (cont-d)

- Thus:
  - the inaccuracy $\Delta S_i(t)$ in knowing $S_i(t)$ – i.e., the difference between the approximate and actual values of this probability
  - affects the inaccuracy $\Delta S(t)$ in the following way:

$$\Delta S(t) = \Delta S_i(t) \cdot (P_i^+(t) - P_i^-(t)).$$

- The difference $P_i^+(t) - P_i^-(t)$ between two probabilities always lies in the interval $[-1, 1]$ – and it can be equal to 1 and -1.
- Thus, a priori:
  - before we start simulations and thus, before we gain some information about the conditional probabilities $P_i^+(t)$ and $P_i^-(t)$,
  - the only thing we know about this relation is that

$$|\Delta S(t)| \leq |\Delta S_i(t)|.$$

13. How inaccuracy of knowing $S_i(t)$ affects the inaccuracy of estimating $S(t)$ (cont-d)

- So, to minimize the effect of the uncertainty $\Delta S_i(t)$ on the inaccuracy of estimating $\Delta S(t)$, it is desirable to minimize $|\Delta S_i(t)|$.
- Each of the system's components i can affect the estimate of $S(t)$.
- All these effects should be as small as possible.
- So a natural idea it to minimize the largest of these effect, i.e., to minimize the value $\max(|\Delta S_1(t)|, |\Delta S_1(t)|, \ldots)$.
- If we restrict this value of some small different δ, then:
  - since failures of different components are usually independent events,
  - the overall effect on $S(t)$ will be equal to $\sqrt{C} \cdot \delta$, where C denotes the number of the system's components.

14. How does the inaccuracy of knowing λ_i affect the inaccuracy in $S_i(t)$?

- In this talk, we consider the case when the reliability of each component is described by an exponential distribution.
- In this case, each value $S_i(t)$ is described by a single parameter λ_i.
- So, the inaccuracy in $S_i(t)$ are caused by the inaccuracy of knowing this parameter.
- Suppose that the actual value of this parameter is λ_i, and the approximately value is $\lambda_i + \Delta\lambda_i$ for some small $\Delta\lambda_i$.
- Then, for each moment t, the difference between the corresponding survival functions is equal to

$$\Delta S_i(t) = \exp(-\lambda_i \cdot t) - \exp(-(\lambda_i + \Delta\lambda_i) \cdot t).$$

- This difference is equal to 0 when $t = 0$ and when $t \to \infty$.
- So the largest absolute value of this difference is attained for some moment t_0 between 0 and infinity.

15. How does the inaccuracy of knowing λ_i affect the inaccuracy in $S_i(t)$ (cont-d)

- To find this moment of time t_0, we can differentiate the above expression with respect to t and equate the resulting derivative to 0.
- As a result, we get
$$-\lambda_i \cdot \exp(-\lambda_i \cdot t_0) + (\lambda_i + \Delta\lambda_i) \cdot \exp(-(\lambda_i + \Delta\lambda_i) \cdot t_0) = 0.$$
- If we move the negative term to the right-hand side, we get
$$(\lambda_i + \Delta\lambda_i) \cdot \exp(-(\lambda_i + \Delta\lambda_i) \cdot t_0) = \lambda_i \cdot \exp(-\lambda_i \cdot t_0).$$
- If we divide both sides by $\lambda_i \cdot \exp(-(\lambda_i + \Delta\lambda_i) \cdot t_0)$, we conclude that
$$\frac{\lambda_i + \Delta\lambda_i}{\lambda_i} = \exp(\Delta\lambda_i \cdot t_0), \text{ i.e., } 1 + \frac{\Delta\lambda_i}{\lambda_i} = \exp(\Delta\lambda_i \cdot t_0).$$
- By taking logarithm of both sides, we get
$$\Delta\lambda_i \cdot t_0 = \ln\left(1 + \frac{\Delta\lambda_i}{\lambda_i}\right).$$

16. How does the inaccuracy of knowing λ_i affect the inaccuracy in $S_i(t)$ (cont-d)

- The difference $\Delta\lambda_i$ is small, and for small values v, we have
$$\ln(1+v) \approx v.$$
- Thus, in the first approximation, we have $\Delta\lambda_i \cdot t_0 \approx \dfrac{\Delta\lambda_i}{\lambda_i}$.
- If we divide both sides by $\Delta\lambda_i$, we get $t_0 \approx \dfrac{1}{\lambda_i}$.
- Thus, $\lambda_i \cdot t_0 \approx 1$ and $\exp(-\lambda_i \cdot t_0) = \exp(-1)$.
- Similarly, $\lambda_i + \Delta\lambda_i) \cdot t_0 \approx (\lambda_i + \Delta\lambda_i) \cdot \dfrac{1}{\lambda_i} = 1 + \dfrac{\Delta\lambda_i}{\lambda_i}$.
- Thus,
$$\exp(-(\lambda_i + \Delta\lambda_i) \cdot t_0) = \exp\left(-1 - \frac{\Delta\lambda_i}{\lambda_i}\right) = \exp(-1) \cdot \exp\left(-\frac{\Delta\lambda_i}{\lambda_i}\right).$$
- The value $\Delta\lambda_i$ is small, so we can use the fact that for small value v, we have $\exp(v) \approx 1 + v$.

17. How does the inaccuracy of knowing λ_i affect the inaccuracy in $S_i(t)$ (cont-d)

- Thus, in the first approximation, we have

$$\exp(-(\lambda_i+\Delta\lambda_i)\cdot t_0)=\exp(-1)\cdot\left(1-\frac{\Delta\lambda_i}{\lambda_i}\right).$$

- Thus, the largest value of the difference $\Delta S_i(t)$ – which is attained when $t=t_0$ – is equal to

$$\max_t|\Delta S_i(t)|=\exp(-\lambda_i\cdot t_0)-\exp(-(\lambda_i+\Delta\lambda_i)\cdot t_0)=e^{-1}\cdot\frac{\Delta\lambda_i}{\lambda_i}.$$

- Similar computations show that this difference can be equivalently described as $e^{-1}\cdot\Delta\Lambda_i$, where we denoted $\Lambda_i\stackrel{\text{def}}{=}\ln(\lambda_i)$.

- Thus, in the first approximation, we have

$$\max_t|\Delta S_i(t)|\approx e^{-1}\cdot\Delta\Lambda_i.$$

- So, if the differences between each value Λ_i and its approximation do not exceed some value η, then we have $|\Delta S_i(t)|\le e^{-1}\cdot\eta$.

18. How does the inaccuracy of knowing λ_i affect the inaccuracy in $S_i(t)$ (cont-d)

- Thus, the accuracy of estimating $S(t)$ is bounded by $\sqrt{C}\cdot e^{-1}\cdot \eta$.
- So, if we want to estimate $S(t)$ with some accuracy ε, we need to make sure that $\sqrt{C}\cdot e^{-1}\cdot \eta\le \varepsilon$, i.e., that $\eta\le \eta_0\stackrel{\text{def}}{=}\dfrac{e\cdot\varepsilon}{\sqrt{C}}$.
- Thus, we arrive at the following optimization problem.

19. The resulting optimization problem

- *Given:*
  - the values $\lambda_1, \lambda_2, \ldots, \lambda_C$ describing the reliability of different components, and
  - the accuracy $\varepsilon > 0$ which we want to estimate the system's survival function $S(t)$.
- *Find:* of all the divisions of components into clusters and of selecting the values $\widetilde{\lambda}_j$ for each cluster,
  - for which for each component i within each cluster j, we have $\left|\Lambda_i - \widetilde{\Lambda}_j\right| \le \eta_0$,
  - we select the division with the smallest possible number of clusters.
- Here, $\Lambda_i \stackrel{\text{def}}{=} \ln(\lambda_i)$, $\widetilde{\Lambda}_j \stackrel{\text{def}}{=} \ln\left(\widetilde{\lambda}_j\right)$, and $\eta_0 \stackrel{\text{def}}{=} \dfrac{e \cdot \varepsilon}{\sqrt{C}}$.

20. Solution to the resulting optimization problem: algorithm

- First, we sort the values λ_i and compute their logarithms $\Lambda_i = \ln(\lambda_i)$, so that $\Lambda_1 \le \Lambda_2 \le \ldots \le \Lambda_C$.
- Then, we group all the components for which $\Lambda_i \le \Lambda_1 + 2\eta_0$ into the first cluster.
- For this cluster, we take $\widetilde{\Lambda}_1 = \Lambda_1 + \eta$.
- After that, we take the smallest of not-yet-assigned values Λ_{i_0}, and form a new cluster by all the components i for which $\Lambda_i \le \Lambda_{i_0} + 2\eta_0$.
- For this cluster, we take $\widetilde{\Lambda}_j = \Lambda_{i_0} + \eta$.
- We repeat this procedure until all the components are clustered.

21. What is the computational complexity of this algorithm?

- The time needed to compute C logarithms is proportional to the number of components, so this time is linear in C.
- In computer science notations, the time is $O(C)$.
- Sorting requires time $O(C \cdot \log_2(C))$.
- Following computations deal with each of C components at most twice, so they also take time $O(C)$.
- Thus, the overall computation time is

$$O(C) + O(C \cdot \log_2(C)) + O(C) = O(C \cdot \log_2(C)).$$

- So, it is asymptotically the same time as simply sorting the components by their reliability.

22. Numerical example

- Suppose that we have 10 components with value $\Lambda_i = i$, and $\eta_0 = 1.2$. Then:

- Into the first cluster, we group all the components for which $\Lambda_i \le \Lambda_1 + 2\eta_0 = 1 + 2 \cdot 1.2 = 3.4$, i.e., Components 1, 2, and 3.

- For this cluster, we take $\widetilde{\Lambda}_1 = \Lambda_1 + \eta = 1 + 1.2 = 2.2$.

- The smallest not-yet-assigned value is $\Lambda_{i_0} = 4$.

- So we group all the components for which $\Lambda_i \le \Lambda_{i_0} + 2\eta_0 = 4 + 2 \cdot 1.2 = 6.4$, i.e., Components 4, 5, and 6 – into the second cluster.

- For this cluster, we take $\widetilde{\Lambda}_2 = \Lambda_{i_0} + \eta = 4 + 1.2 = 5.2$.

- Now, the smallest not-yet-assigned value is $\Lambda_{i_0} = 7$.

- So, we group all the components for which $\Lambda_i \le \Lambda_{i_0} + 2\eta_0 = 7 + 2 \cdot 1.2 = 9.4$, i.e., Components 7, 8, and 9 – into the third cluster.

- For this cluster, we take $\widetilde{\Lambda}_3 = \Lambda_{i_0} + \eta = 7 + 1.2 = 8.2$.

23. Numerical example (cont-d)

- The only remaining not-yet-assigned component is Component 10.
- So this component forms the fourth cluster.
- For this cluster, according to our algorithm, we take $\widetilde{\Lambda}_4 = \Lambda_{i_0} + \eta = 10 + 1.2 = 11.2$ (although we could as well take $\widetilde{\Lambda}_4 = 10$).

24. Non-uniqueness

- The above numerical example can be also used to illustrate the fact that there can be several different optimal solutions to our problem.
- For example, instead of Starting with Component 1 and going up, we could start with Component 10 and go down.
- This way, we will form clusters $\{8, 9, 10\}$, $\{5, 6, 7\}$, $\{2, 3, 4\}$, and $\{1\}$.

25. How we can prove that our algorithm provides the optimal clustering

- The smallest value Λ_1 must belong to come cluster.
- The more elements in this cluster, the fewer components remain unclustered.
- So, to minimize the overall number of clusters, it is necessary to make sure that this cluster has as many elements as possible.
- The larger the value $\widetilde{\Lambda}_1$, the more components we can cover by the corresponding cluster – since:
  - these are all components for which $\Lambda_i \le \widetilde{\Lambda}_1 + \eta$, and
  - the larger $\widetilde{\Lambda}_1$, the less restrictive is this inequality and thus, the more components satisfy it.
- The only limitation on the value $\widetilde{\Lambda}_1$ is that this value should be not too far away from Λ_1, i.e., that $\widetilde{\Lambda}_1 \le \Lambda_1 + \eta$.
- The largest value that satisfies this condition is the value $\widetilde{\Lambda}_1 = \Lambda_1 + \eta$.

26. How to prove that our algorithm provides the optimal clustering (cont-d)

- For this value, the condition $\Lambda_i \le \widetilde{\Lambda}_1 + \eta$ takes the form $\Lambda_i \le \Lambda_1 + 2\eta$.
- Thus, the first cluster provided by our algorithm is indeed optimal.
- Once the first cluster is formed, we can apply similar arguments to prove that the second cluster is also optimal, etc.

27. Conclusions

- When we design a complex engineering system, one of the important requirement is that:
  - this system should be sufficient reliable,
  - e.g., that with a given close-to-100% certainty it should function well during a given (sufficiently long) period of time.
- The usual way to gauge the system's reliability on the design stage is by simulations.
- Simulating a complex system usually requires a significant amount of computation time.
- Moreover, we usually need to repeat these simulations several times:
  - If the current design does not lead to the desired reliability,
  - then we need to change the design: e.g., replace some components with more reliable ones or add duplication, and
  - we need to re-estimate the new design's reliability.

28. Conclusions (cont-d)

- As a result, this process often requires an unrealistically long computation time.
- We can speed up computations if the system has several identical components.
- Then, instead of simulating their faults one by one, we can simply simulate the overall number of faulty components.
- This idea can be also used if some components are similar:
  - to speed up computations,
  - we can cluster these components into a single granule and replace the original characteristics of each component by the average-over-this-granule.
- We somewhat modified the components' characteristics.
- So, the resulting estimate of the system's reliability is, in general, somewhat different from its actual reliability.

29. Conclusions (cont-d)

- We want this estimate to be as accurate as possible.
- So, we arrive at the following question:
  - within the given restrictions on available computation time,
  - what division into granules leads to the most accurate estimate of the system's reliability?
- In this talk, we provide an answer to this optimization question.
- Namely:
  - we describe a feasible algorithm for dividing components into granules
  - that leads to the most accurate estimate of the system's reliability.

30. Acknowledgments

This work was supported in part:

- by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329,
- by the US National Science Foundation grants:
    - 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science),
    - HRD-1834620 and HRD-2034030 (CAHSI Includes),
    - EAR-2225395 (Center for Collective Impact in Earthquake Science C-CIES),
- and by the AT&T Fellowship in Information Technology.