Mat-2.4177 Seminar on Case Studies in Operations Research

Tools for analyzing epistemic uncertainties in probabilistic risk analysis

Interim report

Client: VTT (Technical Research Centre of Finland)

Project team:
Janne Laitonen (Project manager)
Henri Losoi
Markus Losoi
Kimi Ylilammi

Version 1	To the team members, Holmberg and Toppila for comments, 2.4.2013
Version 2	Comments included, to the course assistant and opponent team, 3.4.2013
Version 3	Comments by Salo and the opponents included, 10.4.2013
	Approved by VTT, 11.4.2013

1. Introduction

In this project, which is carried out under Aalto University's course Mat-2.4177 Seminar on Case Studies in Operations Research, a method for analyzing the epistemic uncertainties in probabilistic risk analysis is implemented using interval probabilities. The client of the project is VTT (Technical Research Centre of Finland) and the project is linked to a joint project PRADA (PRA Development and Application) by VTT and Aalto University which is carried out under the National Nuclear Power Plant Safety Research Programme 2011-2014 (SAFIR2014).

The method is based on the framework presented by Toppila and Salo [1] where the uncertainty about the basic event probabilities is expressed by intervals. These intervals indicate the knowledge on the true values of the model parameters and contain the belief about the plausible range in which the probabilities may locate. Since there is epistemic uncertainty in the event probabilities, the risk importance measures are impacted by this uncertainty. In the method, dominance relations for the importance measures are established. One event is said to dominate another if its risk importance measure is at least as high for all event probabilities that are within their respective intervals and strictly higher for some probabilities.

This interim report summarizes the progress that the project team has made so far, describes the changes to the initial project plan, and contains a revised risk management plan.

2. Updated objectives and scope

Considerable changes in the objectives or the scope of the project have not been made. The original objectives given in the project plan are still valid:

- Get an understanding about epistemic uncertainty and its modeling with interval probabilities.
- Implement an open source Matlab-package compatible with Finnish PRA software FinPSA for analyzing epistemic uncertainties expressed though interval probabilities.
- Develop the software following good software development practices including the following documents: requirements specification, software design specification, testing plan and quality assurance plan. Software validation report including test results must be produced in the end.

However, due to some setbacks with the implementation concerning considerably long computation times, the team may have to decrease the comprehensiveness of the software testing. As originally planned, the test cases still need to give strong confidence that the software calculates the dominance relations correctly. But the tests on how the model complexity affects the calculation time may be irrelevant due to the long computation time. Thus, the test cases may not include as large and complex models as originally planned.

3. Current status of the tasks and the project

So far, the project has developed well although some tasks have required more time and effort than originally planned leading to delays. Software requirements specification (task 4, see the table below) and quality assurance plan (task 5) as well as software testing plan (task 8) are in good phase and the scope of the documents is considered acceptable but they still need to be completed and finalized.

Currently, the implemented software successfully calculates simple test cases and the cases provided in [1]. This was set as an internal goal in the project plan. However, there are delays in the implementation (task 6) and the software design documentation (task 7). For instance, the software still needs more rigorous commenting and harmonizing the coding practices. The quality of the code needs to be improved and some resources need to be addressed on how to reduce the computation time (e.g., using Symbolic Math Toolbox) since for now the computation time is out of practical purposes, e.g., a test case with 147 minimal cut sets [1] takes approximately four hours. The delays are mainly due to underestimated resources needed for the implementation and heavy workload for the team (see risk scenario 2 in section 5) in terms of work and midterms.

The status of each task defined in the project plan is described below. The colors on the left correspond the status of the task: Green corresponds to completed, orange is under work and red indicates that the task has not been started.

	Task 1: Understanding the backgrounds and the theory **Status:** Finished as a task although continuous learning takes place during the project.
	Task 2: Define the objectives of the project **Status:** Finished in conjunction with the project plan.
	Task 3: Project plan and presentation **Status:** Finished. The final version was approved by VTT 22nd of March.
	Tasks 4 and 5: Requirement specification and quality assurance plan **Status:** Second drafts written, the comments received from VTT and Toppila to be implemented. The scope of the documents is acceptable but the documents need to be completed and finalized.
	Task 6: Implementation of Matlab-package **Status:** The implementation is under work and more resources have been allocated for this task due to delays. The current version successfully calculates the cases taken from [1] and the interface with FinPSA is almost complete. The software still requires efforts in commenting and harmonizing the coding practices as well as improving the quality of the code. In addition, some resources need to be addressed on how to reduce the computation time, e.g., using Symbolic Math Toolbox.
	Task 7: Software design specification **Status:** This task has been delayed due to difficulties faced with the software implementation and needs to be started as soon as the software design is ready enough to be documented.
	Task 8: Testing plan **Status:** First draft written, the comments received from VTT and Toppila to be implemented. The scope of the document is acceptable but the document needs to be completed and finalized. Consistency with the quality assurance plan needs to be assured.
	Task 9: Interim report and presentation **Status:** Ready by 4th of April. Finalization needed according the comments received from the seminar.
	Task 10: Software testing and reporting of the test results **Status:** The current version of the software has been successfully tested using simple test cases (i.e., using cases A+B, A+B+C+D and A+BC) and the cases provided in [1]. The final testing is to be done during week 17. The test cases need to be documented.
	Task 11: Final report and presentation **Status:** To be done during weeks 17 and 18.
	Task 12: Finalization **Status:** To be done after the final seminar.
	Task 13: Comments for opponent team **Status:** To be done before the seminars.

4. Re-allocation of tasks and schedule

Due to the problems faced with the software implementation, the resources have been re-allocated in order to ensure the fulfillment of the corresponding tasks. Therefore, Markus and Kimi will concentrate mostly on software development and documentation. For example, separating the slowest part of the algorithm into separate functions is believed to help the further development of the software. In addition, Henri will focus on how to reduce the computation time using Symbolic Math Toolbox which could be implemented to replace the separated slow functions. During the project the team has also observed that in practice the code developers test the early versions of the software. Therefore, it was considered practical that Markus and Kimi test the software during the development phase and finally Henri will test the complete version.

The re-allocation means that the team may have to decrease the comprehensiveness of the software testing and the test cases may not include as large and complex models as originally planned. In addition, Janne needs to take more responsibility on reporting, software requirement

specification and possibly on testing plan. The final report and presentation will be prepared by the whole team. The responsibilities are allocated as follows:

Janne:	Interim report and presentation, quality assurance plan, software requirement specification, producing test cases with FinPSA, project management
Henri:	Reducing the computation time with Symbolic Math Toolbox, testing plan and final testing
Markus:	Further implementation of the algorithm to calculate dominance relations, software documentation and testing during the development phase
Kimi:	Further development of the interface with FinPSA, software documentation and testing during the development phase

The schedule, important dates and the tasks:

Weeks 14 to 15:	Interim report and presentation, quality assurance plan, software development and documentation, Symbolic Math Toolbox.
End of week 15:	Go/no go –decision on whether Symbolic Math Toolbox will be utilized. Possible implementation during week 16. Quality assurance plan complete (excluding the finalization during the last week).
End of week 16:	Complete version of the software. Software requirement specification and testing plan complete (excluding the finalization during the last week).
End of week 17:	Final testing passed. Software design specification complete (excluding the finalization during the last week).
Weeks 17 to 18	Writing the final report.
Week 19:	Finalization of the report and other documents. The results are presented in the final seminar on Friday 10th of May.
Week 20:	Finalization of the project based on the comments received in the seminar and from VTT.

Future meetings with VTT, time and agenda:

9th of April:	Demonstration of the software
Week 17:	Status of the software and the software design documentation, test plan and testing
Week 19:	Drafts of the documents and the final report, test results
Week 20:	Possible meeting to close the project, demonstration of the software and delivering the results of the project

5. Revised risk management plan

The updated risks and the revised risk management plan are shown in the table below. As stated before, risk scenario 2 can be considered to become true, at least partly. Since the schedule is now not as flexible as in the beginning of the project delays may have bigger influence on the project. Due to these reasons the likelihood of the scenario 2 is increased from medium to high and the importance of the consequence is considered high instead of medium. Instead, the likelihood of the scenario 3 is decreased from medium to medium or low due to the fact that at this point of the project the team and the stakeholders have a better idea on the project requirements. Scenarios 7 and 8 are new and originate from the delays and setbacks described before. According to the risk management plan, scenarios 2, 7 and 8 pose the biggest risks.

Scenario 1: One or several team members catch a cold or fall ill. **Likelihood:** High (already observed twice) **Consequence:** From minor delays to partial project failure depending on the duration, timing and number of members that are unavailable. Importance: from low to high. **Management**: Sharing information and documents, prioritization of tasks. SVN repository is to be used for the software versioning. Google Drive folder is used for saving and sharing the project documentation. Ideas and know-how is shared in regular meetings. If necessary, higher rank tasks are prioritized and the less important are omitted.

Scenario 2: Workload is too heavy for the team members. **Likelihood:** High (the project manager studies part-time and other members have many courses to complete) **Consequence:** Delays and possible reductions on the documentation quality. Importance: High. **Management:** The group has meetings on regular basis where possible delays and problems can be discussed. If needed, tasks can be re-allocated. The team also considered important not to start widen the scope of the project but to keep the required goal in mind. If absolutely necessary, the project manager can ask some days off from the daywork.
Scenario 3: Project requirements are unclear for the team due to several stakeholders, i.e., VTT, Aalto University and Finnish Radiation and Nuclear Safety Authority (STUK). **Likelihood:** Medium or low (the requirements should be clearer for everyone by now) **Consequence:** Confusion may lead to delays or different opinions on the success of the project. Importance: medium. **Management:** This ambiguity was discussed with Salo on 8th Feb 2013 and it was confirmed that VTT sets the objectives and requirements for the project. Nevertheless, some ambiguity may still remain and therefore the team will communicate with Holmberg regularly.
Scenario 4: The method is not applicable for solving the dominance relations. **Likelihood:** Low (the method has gone through a journal review process) **Consequence:** The software might be inapplicable but the success and the quality of the requirements specification, test plan etc. are not dependent on that. Moreover, if some deficiencies were to be found in the method it would not lower but more likely increase the value of the project. Importance: Low. **Management:** Due to low likelihood and low importance, no specific mitigation method is planned. If needed though, the problems can be discussed with Holmberg, Toppila and Salo.
Scenario 5: The data for testing is not applicable or available. **Likelihood:** Low (Toppila has already provided the data he used for testing. In addition, Laitonen can use FinPSA-software to produce test-cases.) **Consequence:** If the data for testing is not available, the implemented software cannot be tested extensively. Importance: Medium. **Management:** Since the team already has received some data and results for comparison (paper by Toppila and Salo) this scenario is considered highly unlikely.
Scenario 6: Conflicts among team members. **Likelihood:** Low (Most of the team members know each other beforehand and it seems that the team works well together.) **Consequence:** Decrease in efficiency and motivation possibly leading to delays and reduction on the quality of the documentation. Importance: Medium. **Management:** Fair allocation of tasks among team members. Efforts to keep up positive and optimistic (yet realistic) attitude.
Scenario 7: The implemented software is too slow for practical purposes. This scenario originates from the scenario 4. **Likelihood:** High (so far this is the case) **Consequence:** The software cannot be used in practice and needs further development by VTT. Importance: High. **Management:** The resources have been re-allocated in order to lower the risk and an alternative method to speed up the computation is under study. More information is provided in the previous sections.
Scenario 8: Testing may not be as comprehensive as originally planned. **Likelihood:** High (due to the delays and setbacks this may become true) **Consequence:** The test cases need to give strong confidence that the software calculates the dominance relations correctly but the test cases may not include as large and complex models as originally planned due to long computation time. Strictly speaking though, this does not conflict with the requirements of the project. Importance: medium. **Management:** As for the scenario 7.

References

[1] Toppila A, Salo A. Prioritization of events in fault tree analysis under interval-valued probabilities. Manuscript, 2013.