UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Improving the process of analysis and comparison of results in dependability benchmarking for computer systems

PhD thesis

Author: *Miquel Martínez Raga*

Supervisors: *David de Andrés Martínez*
Juan Carlos Ruiz García

October 4, 2018

- A la meva família -

Acknowledgments. Agraïments

La meua àvia em deia quan era xicotet allò de *"de bien nacido es ser agradecido"*, i jo sóc dels que pensa que quan algú t'ajuda, o et dóna suport, això s'ha d'agrair. Així doncs, aquest espai el dedique per a mostrar el meu agraïment a aquelles persones que, d'una manera o d'altra, han estat amb mi durant el desenvolupament d'aquesta tesi i m'han ajudat a arribar fins ací.

Començaré dirigint-me als meus directors, als quals considere amics, i que han sigut una peça clau en aquest procés, ja que si no fos per ells no m'hagués embarcat en aquesta aventura.

A Juan Carlos, porque contigo empezó todo. Tú supiste ver más allá, y así pasamos del PFC al Máster, y luego al doctorado. Y porque sin esas charlas tecnológicas a la hora del café o sin esa actitud tuya que siempre me ha empujado a hacer más, esto no habría sido posible. Por eso, y más, gracias.

A David, porque eres un *crack*, así de sencillo. Porque gran parte de este trabajo lo hemos discutido de la mejor manera, entre almuerzos y risas, y eso me ha hecho disfrutar de estos años de trabajo. Y porque he de reconocer que he aprendido mucho de ti, y no sólo a nivel técnico, sino también como persona. Muchas gracias por todo.

Voldria donar les gràcies a la meua família, perquè sempre m'heu donat suport i d'una manera o d'altra heu format part de les meues experiències. Començant per les últimes incorporacions, que sempre aconsegueixen alegrar-me el dia, fins als més majors, dels que he aprés grans valors per a la vida. I un agraïment especial als meus pares, Elvira i Miguel, no només pel suport i estima constant que sempre he rebut per part vostra, també per tots els esforços que sé que heu fet (i aquells que desconec) per mi, i sense els quals

no hagués arribat fins ací. I perquè si algú m'ha ensenyat el que és ser fort i superar les adversitats, heu sigut vosaltres dos. Vos estime molt. Moltes gràcies.

A les meues amigues Luz i Carme, la meua segona família, per estar sempre que ha fet falta, i el més important, per demostrar-me cada dia la vostra confiança en mi. Vosaltres heu sigut instigadores de què emprengués noves experiències i nous reptes. Per això, per la vostra amistat, i per tota la resta, moltíssimes gràcies a les dues.

Als meus amics i doctors de l'autodenominat *GH team*. Pels bons moments que hem passat junts (i pels dolents també, clar que sí), i perquè m'he alimentat de la vostra experiència, heu obert camí i heu sabut fer-me veure les parts bones del doctorat quan tot estava molt negre. Escamilla, Flores, Jaume, Joan i Mario, sou tots uns màquines i us estic molt agraït per tot.

Als meus amics que m'han donat el seu suport en aquesta etapa de la meua vida. A Franchu i a Ousé, que tot i estar lluny, sempre haveu tingut les males paraules adequades per a inspirar-me, i les bones quan les he necessitades, moltes gràcies als dos. A Antonio, te quiero dar las gracias por tu apoyo durante estos años, por nuestros debates y discusiones que siempre me han dado una "vidilla" que es difícil de describir, muchas gracias amigo. A mi amiga Cristina, tu que has vivido de primera mano mi día a día, me has dado tu apoyo y me has contagiado tu buen humor y fuerza para luchar durante este tiempo, muchas gracias.

Agradecer también a los miembros del grupo de investigación del que he formado parte durante todo este tiempo, el *Grupo de Sistemas Tolerantes a Fallos* (GSTF). Con vosotros he aprendido lo sagrado que pueden llegar a ser los almuerzos de los viernes a las 10, y todo porque desde mis comienzos como becario me habéis tratado como uno más del grupo. Muchas gracias a todos.

I would like to dedicate a special thanks to the people that became part of my life in Coimbra. Thanks to everyone in Amoreira's house, you guys really made me feel like home, and that meant everything to me. Thanks to the guys I worked with at the CISUC, you guys helped me out a lot, even to improve my *"Portuñol"*! In particular, I want to thank Marco Vieira for giving me the opportunity to work with him and his team, and thank Nuno Antunes, who made my work there easier with his sense of humor and his friendship, thank you all.

Em deixe a molts amics i amigues per mencionar, però no els oblide, i vull concloure agraint a totes aquelles persones que, d'una manera o altra, conscientment o inconscientment, m'han acompanyat en el camí que he seguit fins a arribar a aquest punt. A tots i totes, de tot cor, moltíssimes gràcies.

Abstract

Dependability benchmarks are designed to assess, by quantifying through quantitative performance and dependability attributes, the behavior of systems in presence of faults. In this type of benchmarks, where systems are assessed in presence of perturbations, not being able to select the most suitable system may have serious implications (economical, reputation or even lost of lives). For that reason, dependability benchmarks are expected to meet certain properties, such as *non-intrusiveness*, *representativeness*, *repeatability* or *reproducibility*, that guarantee the robustness and accuracy of their process. However, despite the importance that comparing systems or components has, there is a problem present in the field of dependability benchmarking regarding the analysis and comparison of results.

While the main focus in this field of research has been on developing and improving experimental procedures to obtain the required measures in presence of faults, the processes involving the analysis and comparison of results were mostly unattended. This has caused many works in this field to analyze and compare results of different systems in an ambiguous way, as the process followed in the analysis is based on argumentation, or not even present. Hence, under these circumstances, benchmark users will have it difficult to use these benchmarks and compare their results with those from others. Therefore extending the application of these dependability benchmarks and perform cross-exploitation of results among works is not likely to happen.

This thesis has focused on developing a methodology to assist dependability benchmark performers to tackle the problems present in the analysis and comparison of results of dependability benchmarks. Designed to guarantee the fulfillment of dependability benchmark's properties, this methodology seamlessly integrates the process of analysis of re-

sults within the procedural flow of a dependability benchmark. Inspired on procedures taken from the field of operational research, this methodology provides evaluators with the means not only to make their process of analysis explicit to anyone, but also more representative for the context being.

The results obtained from the application of this methodology to several case studies in different domains, will show the actual contributions of this work to **improving the process of analysis and comparison of results in dependability benchmarking for computer systems**.

Resumen

Los *dependability benchmarks* (o benchmarks de confiabilidad en español), están diseñados para evaluar, mediante la categorización cuantitativa de atributos de confiabilidad y prestaciones, el comportamiento de sistemas en presencia de fallos. En este tipo de benchmarks, donde los sistemas se evalúan en presencia de perturbaciones, no ser capaces de elegir el sistema que mejor se adapta a nuestras necesidades puede, en ocasiones, conllevar graves consecuencias (económicas, de reputación, o incluso de pérdida de vidas). Por esa razón, estos benchmarks deben cumplir ciertas propiedades, como son la *no-intrusión*, la *representatividad*, la *repetibilidad* o la *reproducibilidad*, que garantizan la robustez y precisión de sus procesos. Sin embargo, a pesar de la importancia que tiene la comparación de sistemas o componentes, existe un problema en el ámbito del *dependability benchmarking* relacionado con el análisis y la comparación de resultados.

Mientras que el principal foco de investigación se ha centrado en el desarrollo y la mejora de procesos para obtener medidas en presencia de fallos, los aspectos relacionados con el análisis y la comparación de resultados quedaron mayormente desatendidos. Esto ha dado lugar a diversos trabajos en este ámbito donde el proceso de análisis y la comparación de resultados entre sistemas se realiza de forma ambigua, mediante argumentación, o ni siquiera queda reflejado. Bajo estas circunstancias, a los usuarios de los benchmarks se les presenta una dificultad a la hora de utilizar estos benchmarks y comparar sus resultados con los obtenidos por otros usuarios. Por tanto, extender la aplicación de los benchmarks de confiabilidad y realizar la explotación cruzada de resultados es una tarea actualmente poco viable.

Esta tesis se ha centrado en el desarrollo de una metodología para dar soporte a los desarrolladores y usuarios de benchmarks de confiabilidad a la hora de afrontar los pro-

blemas existentes en el análisis y comparación de resultados. Diseñada para asegurar el cumplimiento de las propiedades de estos benchmarks, la metodología integra el proceso de análisis de resultados en el flujo procedimental de los benchmarks de confiabilidad. Inspirada en procedimientos propios del ámbito de la *investigación operativa*, esta metodología proporciona a los evaluadores los medios necesarios para hacer su proceso de análisis explícito, y más representativo para el contexto dado.

Los resultados obtenidos de aplicar esta metodología en varios casos de estudio de distintos dominios de aplicación, mostrará las contribuciones de este trabajo a **mejorar el proceso de análisis y comparación de resultados en procesos de evaluación de la confiabilidad para sistemas basados en computador**.

Resum

Els *dependability benchmarks* (o benchmarks de confiabilitat, en valencià), són dissenyats per avaluar, mitjançant la categorització quantitativa d'atributs de confiabilitat i prestacions, el comportament de sistemes en presència de fallades. En aquest tipus de benchmarks, on els sistemes són avaluats en presència de pertorbacions, el no ser capaços de triar el sistema que millor s'adapta a les nostres necessitats pot tenir, de vegades, greus conseqüències (econòmiques, de reputació, o fins i tot pèrdua de vides). Per aquesta raó, aquests benchmarks han de complir certes propietats, com són la *no-intrusió*, la *representativitat*, la *repetibilitat* o la *reproductibilitat*, que garanteixen la robustesa i precisió dels seus processos. Així i tot, malgrat la importància que té la comparació de sistemes o components, existeix un problema a l'àmbit del *dependability benchmarking* relacionat amb l'anàlisi i la comparació de resultats.

Mentre que el principal focus d'investigació s'ha centrat en el desenvolupament i la millora de processos per a obtenir mesures en presència de fallades, aquells aspectes relacionats amb l'anàlisi i la comparació de resultats es van desatendre majoritàriament. Açò ha donat lloc a diversos treballs en aquest àmbit on els processos d'anàlisi i comparació es realitzen de forma ambigua, mitjançant argumentació, o ni tan sols queden reflectits. Sota aquestes circumstàncies, als usuaris dels benchmarks se'ls presenta una dificultat a l'hora d'utilitzar aquests benchmarks i comparar els seus resultats amb els obtinguts per altres usuaris. Per tant, estendre l'aplicació dels benchmarks de confiabilitat i realitzar l'explotació creuada de resultats és una tasca actualment poc viable.

Aquesta tesi s'ha centrat en el desenvolupament d'una metodologia per a donar suport als desenvolupadors i usuaris de benchmarks de confiabilitat a l'hora d'afrontar els problemes existents a l'anàlisi i comparació de resultats. Dissenyada per a assegurar

el compliment de les propietats d'aquests benchmarks, la metodologia integra el procés d'anàlisi de resultats en el flux procedimental dels benchmarks de confiabilitat. Inspirada en procediments propis de l'àmbit de la *investigació operativa*, aquesta metodologia proporciona als avaluadors els mitjans necessaris per a fer el seu procés d'anàlisi explícit, i més representatiu per al context donat.

Els resultats obtinguts d'aplicar aquesta metodologia en diversos casos d'estudi de distints dominis d'aplicació, mostrarà les contribucions d'aquest treball a **millorar el procés d'anàlisi i comparació de resultats en processos d'avaluació de la confiabilitat per a sistemes basats en computador**.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The rapid and constant evolution of technology in the past decades has made possible the integration of computer-based systems in basically any kind of product, making them present in almost every domain of people's lives. Actually, the growing number of components, the reduction of integration scales and the improvements in connectivity technologies, let developers to keep surprising people with new products able to connect and share information among them. Indeed, it is expected that in a near future, cities will become *smart* [122], and almost every aspect of people's daily lives will be connected to the *Internet of Things* (IoT) in one way or another [12].

In this scenario, the use of *off-the-shelf* (OTS) systems, commercial (COTS) or not, and the reuse of software and hardware components help manufacturers reduce costs in the development process and benefit from the rapid integration of novelties and new functionalities. Thus, providing added value and reducing their time-to-market, allowing them to be more competitive. Therefore, given the amount of alternatives available, it is necessary to define procedures that enable their comparison in order to select the alternative, or the configuration providing the best performance, the best standards or the best features for the product.

In this line of work, non-profit organizations with top quality companies (AMD, CISCO, DELL, etc.) as members, like the *Embedded Microprocessor Benchmark Consortium* (EEMBC) [38], the *Standard Performance Evaluation Corporation* (SPEC) [96] or the *Transaction Processing Performance Council* (TPC) [107], have been devoted to the definition of procedures known as *benchmarks*.

A benchmark is a standardized procedure defined to assess the performance of components or systems under particular conditions. It can be a program, or a set of them, designed to assess systems under specific conditions while performing measurements on them to quantify their performance, enabling end users to compare them.

Currently, the reuse of software and hardware components has also become a common practice in *critical* systems. The type of systems that fall into the critical domain are those where a failure of the system may have severe side-effects. In *safety-critical* systems failures may affect human lives or endanger the environment, while in *business-critical* systems, the impact of a failure will be related with great economical and reputation loses.In *mission-critical* systems, where there may be no way to access the system to fix failures or faults after a mission has started (as it happens when a satellite is placed in orbit), systems must be built to last. Hence, these computer-based systems must be designed and developed to mitigate the effect of accidental faults and tolerate them if possible. To do so, and avoid the occurrence of failures after systems are built, it is necessary to assess these systems in presence of faults while they are still under development.

The work done in [11] defined the standards for the development of fault injection procedures under controlled circumstances. These procedures allowed the evaluation, during the development phase, of the robustness of systems in the presence of faults, and even more important, they served as means for developers to validate developed *fault-tolerant* mechanisms. Nevertheless, in order to reuse existing components, it should be necessary not only mechanisms to assess such components in the presence of faults, but also procedures to compare such assessments, determining which are more suitable to increase a system's robustness. Despite a wishful thinking that traditional benchmarks could be used in conjunction with fault injection procedures, the truth is that these kind of benchmarks were not designed for this purpose, thus making necessary the definition of new procedures.

To cope with this need, the *Dependability Benchmarking* project (DBench) [30] provided a specification on how the procedures for benchmarking systems in the presence of faults should be defined, assessing not only performance attributes, but also dependability and security ones. The DBench project proposes a validation approach for these procedures, known as **dependability benchmarks**, that consists in the verification of a set of key properties that any dependability benchmark should meet to be meaningful under economically acceptable conditions. These properties guarantee that a benchmark

is **representative** of a context of use, **repeatable and reproducible** by anyone, **portable** to other target systems and **non-intrusive** during the evaluation, to mention a few.

The work done in the DBench project set the ground for many works focused on the development of well-defined dependability benchmarks in multiple application domains [67]. In addition to such works, others have put their efforts on improving the quality of the measurements taken during the evaluation of the system, used to quantify performance and dependability attributes. The work done in [17] stresses not only the need of selecting the proper measures to characterize the system, but also the importance of using instruments and tools able to perform accurate measurements with a low level of uncertainty.

Nevertheless, despite all these efforts to guarantee the compliance of dependability benchmarks with the properties specified in the DBench, there is still an important phase of dependability benchmarks that has not been properly studied, the *analysis of results*. Its importance relies on the fact that it constitutes the final step in a dependability benchmark, where measures that categorize the system must be interpreted, easing the comparison amongst alternatives.

But despite the relevance of this phase in the comparison process, it did not received the proper attention by research studies, which were mostly focused in the design and application, or improvement, of dependability benchmarking procedures in multiple domains. Under these circumstances, when comparing a set of alternatives, the analysis of resultant measures must be carried out by the evaluators performing the benchmark, who most likely will be applying their own criteria, and will make use of familiar methodologies. Even though this is not a problem per se, as the evaluator should determine how the analysis of results is carried out, the lack of standardized procedures can introduce bias in the conclusions.

Letting the *analysis of results* in hands of the evaluator, turned out in many works providing an unclear description of the analysis, or not description at all. This situation makes it very difficult (if not impossible) for any evaluator to understand and repeat the process followed to provide the conclusions. This contradicts what is specified in the DBench project, that *dependability benchmarks should be repeatable*, so if the same benchmark is performed again on the same system (or set of systems), and the same environment, results must be statistically equivalent. If the analysis of results is not clear, nor explicit, when comparing the robustness and performance of components to choose among them, *can those results be compared with those from future assessments?*, or instead, *will it be necessary to analyze previous results again?* Even more, let us imagine that an external evaluator A, tries to repeat the dependability benchmark performed on a group of systems by another research group B. Even with statistically equivalent benchmark results, if the process of analysis used by B cannot be repeated, and A follows a different approach, the

conclusions from both works may not be comparable. So, in this scenario, if the same dependability benchmark is applied to assess a new system, *could the conclusions drawn from comparing the new results to those from other evaluators be trusted?* Well, *they should not*, as the analysis followed should be clear enough to be repeated, thus allowing to understand (therefore trust) the reported conclusions.

Unlike what authors did in [17] regarding the properties that measures and measurement tools should have, the main issue is that up to date there are no specifications whatsoever that determine what properties should be satisfied during the analysis of results. So, in this work the properties defined in the DBench project have been *reinterpreted and adapted for the context of the analysis process*. By doing so, it can be seen that such properties are not met in the analysis of results, thus **the compliance of these properties for the whole process of a dependability benchmark can be jeopardized due to the analysis of results**.

Results from different works should not be compared unless they are analyzed following the same procedure, or the differences between both analysis are understood. However, this is not possible if the process of analysis is not made clear and explicit, and therefore repeatable. This problem was already pointed out by [17], where it is stated that *"comparing the results obtained from different approaches to quantitatively assess a system is quite difficult, if not meaningless"*. This also affects the notion of *reproducibility* of a benchmark, where another party implementing the benchmark from its specification to assess the same system, but in a different environment, should obtain statistically equivalent results. It is necessary to follow the same approach during their analysis to compare results from different works, otherwise the conclusions drawn would be meaningless, even if it is assumed that results are statistically equivalent among works. This is why the characterization of the process used to analyze the results should be considered as an important part of a dependability benchmark, as it would contribute to guarantee the comparison and cross-exploitation of results among works. But, these are not the only properties that should be considered in the process of analysis, as there are problems related to the way measures are handled to rank and compare benchmarked systems.

Most performance benchmarks assess the systems according to a single type of measure. For example, the "Autobench" benchmark (from EEMBC [38]) calculates the performance of microprocessors and microcontrollers in *iterations per second*, and the TPC-C benchmark (from TPC [107]), evaluates the performance of *On-Line Transaction Processing* (OLTP) in *transactions per minute*. Since only one measure is used, ranking different systems is straightforward, as it can be done directly. Even when more than one measure is provided by these benchmarks, providing a single score to rank the alternatives is usually straightforward. For example, when measuring the time required to execute several algorithms individually, since these measures are expressed in the same units, like *seconds*, a single score can be obtained easily. However, in addition to perfor-

mance, dependability benchmarks assess robustness, or security attributes of the system, so there exist heterogeneity among the measures used to characterize the behavior of the system. This heterogeneity is caused by the different units and scales used to quantify final measures. Then, unlike what happens in performance benchmarks, common approaches like the arithmetic or geometric mean cannot be applied directly to provide a single score that would let us rank different systems easily.

This problem can be seen in works like [48] and [37], where the analysis of results is not performed, but instead the results obtained are discussed to draw some conclusions from them, and determine which alternatives are better than others. Nevertheless, and even though not tackled in their work, in [37] the authors mention the importance that the application context has on the interpretation of the results, as different application contexts may require to consider some measures more important than others. This issue is usually not addressed in the field of dependability benchmarking, and it refers to another property that should be satisfied by a dependability benchmark, the *representativeness*. In the same way that the workload and the faultload in a dependability benchmark should mimic those that a system would experience in a real scenario, the analysis of results should also mimic the process used to compare alternatives in a real situation. For the analysis to be representative, it is necessary that the requirements imposed by the context can be mapped into it. In order to achieve a representative analysis, evaluators must be able to determine the relative importance among the attributes assessed on the system, indicating which attributes are more relevant to assess the behavior of a system. Otherwise, the analysis will be done considering that all of them are equally important to quantify the behavior of the system. Although that does not mean that this approach is wrong, it is an idealistic approach that does not reflect what happens in real life.

Then, the work done in this thesis has been focused on providing solutions to the problems present in the field of dependability benchmarking. Inspired in the work done in the DBench project [30], this thesis presents a specification of those properties that should be part of the analysis of results for a dependability benchmark. The methodology developed contributes to the integration of the process of analysis in the dependability benchmarking procedure, so the whole process can be considered: *repeatable*, *reproducible*, *representative* and *non-intrusive*. Its structure assures that the analysis remains clear and explicit, and so that every part of the analysis can be verified and validated by anyone. The use of *multi-criteria decision making* (MCDM) techniques, widely applied in different research domains, is proposed to deal with the problem of comparing different alternatives according to a set of heterogeneous measures. At the same time, *back-to-back* testing is used in this work to verify the correctness of the process analysis, detecting (and correcting, when possible) any faults made during its implementation from its definition.

For the sake of comprehension, the following sections will provide insights on the work done in the DBench project regarding dependability benchmarking procedures, and their properties. Additionally, to illustrate why the analysis of results in dependability benchmarking is actually a multi-criteria decision problem, background on these problems, and on the MCDM methods used to solve them is provided. Besides, the main objectives defined in this thesis, as well as the structure of this document, are both also presented in this chapter.

1.2 Dependability Benchmarking

A dependability benchmark is a procedure designed to characterize in a generic and reproducible way the behavior of a computer-based system, or computer components, in the presence of faults. Even though they can be used for comparative purposes, like performance benchmarks, dependability benchmarks have further applications that can benefit both end-users and manufacturers. The characterization of the behavior in the presence of faults can be used to identify vulnerable parts of a system. So, adjustments or improvements in the configuration of the system, or components, can be done to increase robustness. But of course, being able to compare results from different evaluations is crucial, not only to determine which system or component shows a better behavior in the presence of faults, but also to determine if an improvement in a system has been successful.

Then, for dependability benchmarks to be used, it is necessary for their procedures and results to be trusted, or in other words, their use should be accepted by the computer and user communities, as it happens with performance benchmarks. The work done in the Dependability Benchmarking Project (DBench) [30] was focused on achieving this goal by providing means to define dependability benchmarks whose procedures could be verified and validated. A set of guidelines were defined in this project to develop benchmarks that could be considered as *useful*. These guidelines identified the three phases that a dependability benchmark must have, and the procedures that should take part on each phase.

The first phase consists in specifying the dependability benchmark. This specification must be clear enough and unambiguous so it can be used by anyone to implement the dependability benchmark. It can include samples of source code, tools used for the implementation of the benchmark, or anything considered necessary to grant that the benchmark can be implemented from the specification. However, there are some aspects of the benchmarking process whose definition is mandatory in the specification:

- A benchmark can be developed to assess the behavior of a whole system, or it can target a specific part of it. Thus, it is important to make clear what is the *Benchmark Target* (BT).
- Usually, the BT is part of a bigger system on which the experimentation will be conducted, called the *System Under Benchmark* (SUB). If an operating system was the BT, the hardware where it runs (PC, cellphone, smartwatch, etc.) would be the SUB.
- A benchmark can be used with different objectives (identify weak points of a system, make a decision between components, etc.) and by people with different perspectives of the system. The *benchmark context* must be specified to make clear the purpose of the benchmark.
- The set of *measures* provided by the benchmark that will be used to characterize the different aspects of the system (performance, dependability, security, etc.). These measures must be representative for the *context of use* of the benchmark.
- The aspects regarding the experimentation of the benchmark are defined in the *execution profile*, which includes:
  - The *workload* that is executed during the evaluation, which for the benchmark to be useful, will be representative of the application domain and purpose for which the benchmark is defined.
  - The *faultload* of the benchmark, that determines the set of faults injected on the system to emulate the threats that the system would experience in a real situation. As it happens with the workload, representativeness here is key for the benchmark to be useful.
  - The kind of *measurements* that are obtained during the execution of the workload and the faultload, which are later processed to obtain the final measures of the benchmark, also defined in the specification.
  - The *changeload*. Although it is not always required, some type of systems are subjective to changes that must also be considered when performing the experiments. Like it would happen in an ad hoc network where nodes have a dynamic behavior, instead of static, which introduces changes in the topology of the network.

After the dependability benchmark has been implemented from the *specification*, the next phase is the **execution** of the benchmark. In this phase, the actual evaluation of the system takes place, so the workload and faultload are applied according to the specification,

and the corresponding measurements are performed in the system. Sometimes it is required to execute the benchmark in absence of faults (without the faultload) to determine the behavior of the system in normal conditions, commonly known as *golden run*. This type of execution will provide a set of baseline results that can be used for comparison them with those obtained in the presence of faults, hence identifying the actual impact that a fault has on a system.

There is a last phase of the benchmark, where the raw data from the *measurements* performed during the previous phase are processed to quantify the *measures* defined in the specification, the **evaluation** phase. The value of these measures characterizes different aspects of the system's behavior in the presence of faults, and can be used to infer conclusions about the weaknesses of the system, or for comparison purposes between systems or components.

In addition to define guidelines to specify the procedural structure of a dependability benchmark, the DBench project identified a set of key properties that every dependability benchmark should meet. These properties are essential to assess a system's behavior in a meaningful way, while keeping it acceptable under economical conditions. They must be addressed during the whole dependability benchmarking process, and it has to be possible to verify their fulfillment. Verifying these properties represents a big step towards the validation of the benchmark, which will influence the confidence that one can place on the results provided. Without a proper validation of the procedure, there would be a lack of confidence on a dependability benchmark, ultimately damaging its acceptance by the industry and/or research communities. The properties identified in the work done on the DBench project are the following:

- **Representativeness:** Since a benchmark is defined to assess the behavior of a given type of systems on a given application domain, this property concerns all benchmark attributes that may vary depending on the *benchmark context*. The measures defined, and the different *loads* of the experiment (work-, fault- and change-load) used during the benchmark, are strongly related to the benchmark context. Not every possible measure that can be taken on a system is useful to characterize a particular behavior, neither is any possible load that can be executed. So, it is necessary to carefully select, or define, those attributes that will be useful to characterize the behavior of a system. Hence, the *loads* must mimic, as accurately as possible, the load of work, the type of faults and the type of changes that a system would experience in a real situation. Their representativeness is key for the evaluation performed to be meaningful for the application context considered.

- **Repeatability and Reproducibility:** A benchmark can be considered repeatable when it is executed several times under the *same conditions*, and provide statistically equivalent results across executions. Reproducibility, in the other hand, refers

to the situation where a benchmark is implemented by another party, following the *same specification*, and benchmark the same system, yet obtaining statistically equivalent results. Given the comparative nature of benchmarks, these properties are essential for dependability benchmarks to be accepted. If there would not be a statistical equivalence between the results obtained from applying the same benchmark on the same system, no one would trust the results provided by such benchmark.

- **Portability:** To achieve the comparison purposes of a dependability benchmark, it should be possible to deploy such benchmark on different targets. A benchmark can be considered portable when from its specification, it can be implemented for different targets in the same application domain. This property is strongly related to the capability of a dependability benchmark to compare computer systems or components.

- **Non-intrusiveness:** To assess the impact that faults have on the *benchmark target*, the application of the benchmark must not interfere with the operational conditions of the system under benchmark. If a benchmark introduces changes in the structure, or alters the system's behavior, it would be considered intrusive, and the influence of those changes would impact the accuracy of the assessment done on the system.

- **Scalability:** A benchmark must be able to assess systems of different scales. Scaling rules must be given in the specification of the benchmark, as for larger systems, aspects like the workload or the faultload might be adapted. The fact that a benchmark is able to assess systems of different scales, does not mean that comparing the results from systems of different size could be done in a meaningful way.

- **Time and Cost:** The time required to perform a dependability benchmark must be taken into account. The development of benchmarks to perform a thorough and accurate evaluation of a system, usually implies that the more thorough this evaluation is, the more time it requires. But of course, the different level of complexity among systems must be considered, as the more complex a system is, the more time will probably require the execution of the benchmark. Therefore, process automation is important to reduce the benchmarking time as much as possible. A trade-off between time and cost must be sought for dependability benchmarks to consider this property valid. The costs involved in the application of the benchmark have to be controlled, as a dependability benchmark will only be attractive if its contribution is more valuable than its associated implementation cost.

The work done in the DBench project, set precedent for many works devoted to define dependability benchmarks for different kind of systems and application areas [67]. Most

of the works in this field of research were focused on specifying benchmarks that would satisfy the properties defined in the DBench project. Nevertheless, few works aimed their efforts at improving the part of the dependability benchmarking process that involves the analysis of results, and that should ease the comparison among benchmarked systems.

In [74], for example, on-line analytical processing (OLAP) and data warehousing approaches were proposed as a mean to allow the research community to analyze, compare and cross-exploit results from dependability benchmarking experiments. Others like the European project AMBER [84], suggested the definition of a common repository for sharing the experimental data produced by dependability benchmarks. But, the problem of combining measures in a meaningful and repeatable way to characterize a system's behavior, was not addressed by any of these initiatives. Despite the fact that when considering a large number of heterogeneous measures, which is the case of dependability benchmarks, this problem is of major importance.

The lack of standardized procedures to assist evaluators in the aggregation and interpretation of heterogeneous measures resulted in the final user of the benchmark being responsible for the analysis. As a result, users would be approaching the analysis differently, and in some occasions, the approach followed would neither be clear nor explicit, while in others it would be inexistent, and conclusions would be provided throughout a discussion of the results. Due to this lack of information, when results are obtained through multiple heterogeneous measures, it becomes very difficult, if not impossible, to compare different works in a meaningful way.

Therefore, this thesis have been focused on guaranteeing that properties like *repeatability*, *reproducibility*, *representativeness* and *non-intrusiveness* are met also in the process of analysis in dependability benchmarking. Interpreting and comparing heterogeneous results to infer a decision from a set of alternatives is a common problem to almost any field of research. This type of problem, known as *multi-criteria decision* problems, are commonly addressed by researchers with the use of *multi-criteria decision-making* (MCDM) methods. Hence, in this thesis the use of such methods is proposed to deal with the *multi-criteria decision* problem that is the analysis of results in dependability benchmarking.

1.3 Multi-criteria decision-making methods

In everyday situations, people have to make decisions usually involving multiple and often conflicting criteria, where the improvement of one criterion leads to worsening another. To deal with these decisions people apply their judgment and intuition to weight the criteria. Deciding which criteria are more important than others, and to what extent, is conditioned by the prerequisites that the *decision maker* (DM) has about the problem at

stake. But in many decision problems, intuition and judgment cannot be the tools at use, and it is necessary to properly structure the problem, and explicitly define the evaluation of the criteria.

In this sense, a sub-discipline of *operations research*, known as *multi-criteria decision-making* (MCDM) or *multi-criteria decision-analysis* (MCDA), is focused on providing means to solve this kind of problems. Since its origin in the early 1960, many different approaches and methods have been developed and applied in a large number of areas, structuring complex problems to deal with multiple criteria, and provide more informed and better decisions. Their application has become a common practice, and many of these approaches have been implemented into specialized software to assist end users in their decisions [116].

There exist many kinds of decision making problems, and they are classified according to specific features of the problem. A major distinction in their classification is based on whether the solutions are explicitly or implicitly defined. When the solutions are explicitly defined, it means that a finite number of alternatives (benchmark systems/components in the context of this thesis) are provided in order to solve the problem, and these alternatives constitute the set of possible solutions. If the alternatives are not explicitly known, it means that the number of possible solutions is very large, or even infinite, and finding a solution requires to solve a mathematical model. Given the nature of the problem that is being tackled in this thesis, our research has been focused on the first type of problems, where the alternatives are explicitly known from the beginning.

The particular problem that users must face in dependability benchmarking is that they need to make a decision from a set of alternatives based on their results in multiple criteria (measures that constitute the results of the benchmark). However, the presence of multiple criteria usually means that there is not a unique optimal solution, and to find the best possible solution for a problem, it is necessary to incorporate the preferences of the DM to solve it.

The preferences of the DM, usually expressed in terms of *weights*, are used to quantify the importance that each criterion has to achieve a solution for the problem. In some MCDM methods, these preferences are built by the DM during the actual application of the process, which requires constant interaction from the DM throughout the whole process. Other methods require from these preferences to be available before the process of analysis takes place. Since weights are used to quantify the preferences of a DM, the number of possibilities is huge, thus defining them before the analysis takes place, limits the scope of the decision problem to a particular instance from all the universe of possible preferences. This type of MCDM methods, where preferences are required in advance, are the type of methods studied for their application in this thesis, as they also limit the interaction of the DM in the process to a minimum. It will be seen throughout

this document, that this will help preserving the *non-intrusiveness* during the analysis of results.

The MCDM methods that fall into the mentioned categories (explicit number of possible solutions, and preferences provided prior to the analysis), and that are the ones used in this thesis, require from three common elements to solve the problem: *i)* a set of alternatives that need to be compared, *ii)* two or more criteria that determine the quality of the alternatives involved, and *iii)* a set of preferences defined by the DM that determine the course of the decision process. When these three elements are known, the application of a MCDM method assists the DM on ranking the alternatives, sorting them, or identifying the most (or the least) preferred alternatives for the problem [57]. These elements can be represented through what is known as an *evaluation matrix* (EM) [120].

$$EM = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_M \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_N \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{pmatrix} \end{array} \tag{1.1}$$

Equation 1.1 depicts the structure of an EM for a decision making problem involving the results obtained in N criteria by a set of M alternatives. Alternatives are labeled as A_i, where $i = \{1, 2, \cdots, M\}$, while the criteria are represented as C_j, being $j = \{1, 2, \cdots, N\}$. The value obtained by an alternative A_i in a criterion C_j is labeled as x_{ij}. For the context of dependability benchmarking, these values would represent the results obtained from the application of the benchmark to the alternative A_i. Additionally, the preferences of the DM are expressed through a set of weights, which are associated to each criterion present in the problem. This way, a criterion C_j has an associated weight W_j that determines its relative importance with respect to the other criteria in the decision process. These weights quantify the contribution of each criterion to the main solution, and since its contribution is determined in relation to that from the rest, if percentage units are used to quantify the weights, the sum of all weights should add up to 100%.

There is a large variety of methods that can be used to solve MCDM problems though, and each of these methods deals with the decision making process in a different way. This situation creates what is known as the *multi-criteria decision-making paradox* [108]. The differences in the mathematical procedures followed by MCDM methods cause that sometimes, different conclusions are drawn when solving the same problem with different methods. These differences in the conclusions make you wonder *which is the best MCDM to solve the problem?*, and here is where the paradox appears. Deciding which MCDM is better than other requires to compare them based on a set of multiple criteria,

and by definition, this problem is itself a decision-making problem. Since it is necessary to use a MCDM method to determine which MCDM is the best to solve a decision making problem, there is a paradox. Hence, every method claims to provide the best solution, and it is not possible to prove otherwise. Although when it comes to their application, due to their ease-of-use and ease-to-understand (compared to others), some methods have gained more popularity. Some popular methods have been successfully applied in many domains, such as business [24], education [101], or engineering [41] to name a few.

Given the requirements imposed by this work, it is necessary to study which methods are more suitable to *improve the analysis of results in dependability benchmarks and guarantee the benchmarking properties*. Their integration in the dependability benchmarking process for the analysis of results has been successfully tested in this work, and their application in real case studies is shown later on this thesis.

1.4 Objectives of this thesis

The analysis of results in the field of dependability benchmarking has been rarely considered as a subject for research in this field. There is a direct relation between the amount of research done on the specification and development of dependability benchmarks, and the lack of research on how to perform the analysis. This turned out in dependability benchmarks failing to extend the fulfillment of the properties defined in the DBench project to their process of analysis. The problems present in the analysis have a negative impact on the capacity of dependability benchmarks to fulfill one of their fundamental features, **the ability to compare the behavior of systems and components in the presence of faults**.

With these problems in mind, the main objective of this thesis has been to *improve the process of analysis and comparison of results in dependability benchmarks*. The efforts have been focused towards the definition of standardized procedures regarding the specification and application of the analysis of results for dependability benchmarks. In this way, this thesis aims at improving the cross-exploitation and sharing of results among works, from which both academia and industry could benefit. In order to achieve the main goal, a set of sub-objectives have been defined and classified into three categories, regarding the problem being tackled:

1. ***Easing the interpretation of the analysis of results***

   While people with a profile typical from the industry may prefer a single score to compare benchmarked systems, a more academic/research profile will usually prefer to have all the metrics available to study them in detail. It is necessary for the process of analysis to provide means that preserve the interpretation of the results

from different points of view, providing different levels of granularity that can be useful for distinct user profiles.

2. ***Meeting dependability benchmarking properties***

   - Provide means to assure that the process of analysis remains unambiguous and explicit, assuring the *repeatability* and *reproducibility* of the process of analysis in dependability benchmarks. If this objective is achieved, that would mean that a dependability benchmark would satisfy these properties in the whole procedure, from the specification of the benchmark, to the provision of conclusions.

   - Introduce the influence that the benchmark context has in the analysis to make a decision, making the conclusions *representative* for a given context. Analyzing the resultant measures in a meaningful way for the application context of the benchmark will make a difference. Conclusions will be drawn from better informed decisions, as the restrictions (or features) imposed by the context will be considered, therefore making representative conclusions more likely to be accepted by the community.

   - Structure the process of analysis to reduce as much as possible *intrusive* actions of the DM in the conclusions. Defining the analysis once the results are available may cause DMs (willingly or not) to adjust their analysis so they fit their expected conclusions. So, to avoid biased conclusions, it is necessary to redefine the interaction of the DM with the process of analysis, and remove possible sources of bias for conclusions.

3. ***Verifying the proposed methodology***

   - Verification and validation are key for users to agree on the use of a dependability benchmark. Therefore, in the same way dependability benchmarking procedures should be suitable for verification, it is necessary to develop mechanisms to verify the correctness of the process of analysis.

   - Application and testing of our approach through the analysis and comparison of results in different domains. To test the suitability of our approach, it will be necessary to apply it to different benchmarks, and determine its feasibility in comparison to commonly applied methodologies.

1.5 Structure

The work done in this thesis is presented according to the following structure.

- **Chapter 2.** ***Overview of the proposed methodology***
  This chapter presents a big picture of all the work that has been done throughout this thesis. The methodology presented is structured in different parts based on their contributions to solve the problem of analysis of results in dependability benchmarking. Here, the featuring aspects of these parts are introduced in the context of the methodology, and they will be later described in more depth in the rest of the chapters of this thesis.

- **Chapter 3.** ***Analysis of results in Dependability Benchmarking: Can we do better?***
  This chapter describes a first approach to perform the analysis of results in dependability benchmarking with an MCDM method, the *Logic Score of Preferences* (LSP), to achieve the *reproducibility* in the conclusions by making explicit the process of analysis.

- **Chapter 4.** ***Multi-Criteria Analysis of Measures in Benchmarking: Dependability Benchmarking as a Case Study***
  The work presented in this chapter describes how the analysis performed with the LSP method can be characterized to satisfy the requirements from people of both the academia and the industry. It provides insights on how MCDM methods must be integrated in the dependability benchmark process to assure the fulfillment of the key properties described in the DBench project. The application of the approach is described through its instantiation in three different dependability benchmark case studies in the domain of distributed systems.

- **Chapter 5.** ***Gaining confidence on dependability benchmark's conclusions through "back-to-back" testing***
  Since different MCDM methods follow different mathematical procedures that could lead to different conclusions, this chapter introduces an approach to validate the conclusions driven from the application of a MCDM method. Two MCDM methods that share some similarities in their decision process —LSP and the *Analytic Hierarchy Process* (AHP)— are used to provide evaluators with a back-to-back testing approach. This approach would assist such evaluators to double check the conclusions drawn from the analysis and validate their requirements for the analysis.

- **Chapter 6.** ***From Measures to Conclusions using Analytic Hierarchy Process in Dependability Benchmarking***
  The use of the Analytic Hierarchy Process method requires more interaction in the analysis from an evaluator than other methods, which may interfere in the *repeatability and reproducibility* of the analysis in dependability benchmarks. This chapter presents an *assisted pairwise comparison approach* (APCA) developed during this thesis. The APCA limits the interaction of the evaluator to the definition of the requirements for the analysis, avoiding its further interaction in the process of analysis which may endanger the fulfillment of the repeatability and reproducibility by the benchmark.

- **Chapter 7.** ***Assessment of Ad Hoc Routing Protocols for Network Deployments in Disaster Scenarios***
  This chapter describes the work done on a real application of our methodology to a case study in ad hoc network deployments in disaster scenarios. During a disaster situation, cell towers might broke down, leaving some areas of a city isolated, hence people cannot communicate. In this work, the performance and dependability features of three ad hoc routing protocols is evaluated to quantify their robustness to be used in this extreme situation. The methodology proposed in this thesis is applied to provide meaningful conclusions to the application context of the ad hoc network and determine which protocol is the most suitable to be used.

- **Chapter 8.** ***A Multi-criteria Analysis of Benchmark Results With Expert Support for Security Tools***
  The work done during this thesis can be extended to be applied for different domains and contexts of application. This chapter presents an approach, where the specificities of the application context are considered to support the classification and selection of vulnerability/intrusion tools. The approach considers researchers' expertise to define context-aware quality models that determine how relevant the metrics provided from these kinds of tools are for each particular context. These quality models are applied to two different case studies were vulnerability/intrusion detection tools are benchmarked, so that tools can be ranked for each case study and scenario, and it can be determined which tool is the best choice.

- **Chapter 9.** ***Discussion of results***
  The findings and results of this thesis are gathered and discussed in this chapter. The discussion includes the results presented in the previous chapters, but is not limited to them. Some of the results obtained in this thesis, that still remain unpublished, are also discussed.

- **Chapter 10.** ***Conclusions and future work***
  This last chapter concludes this document by reviewing and discussing the achievements and contributions presented in this thesis. Directions for future lines of work are also presented here.

Chapter 2

Overview of the proposed methodology

*The work done in this thesis has been focused from the very beginning on **improving the process of analysis and comparison of results in dependability benchmarking for computer-based systems**. A methodology has been developed to assist benchmark developers with the definition and implementation of an analysis process compliant with the properties expected from a dependability benchmark, as they are defined in the DBench project. The different aspects of this methodology have been designed to tackle specific problems identified in the analysis of results carried out by works in the field of dependability benchmarking. This chapter presents an overview of the work done during this thesis, and how this work has led to the design of a methodology to integrate the analysis of results within the dependability benchmarking procedure.*

2.1 Introduction

In order to seamlessly integrate the analysis of results within the dependability benchmarking process, as it happens in the DBench project, the approach followed in this thesis has structured the process of analysis into different phases. Figure 2.1 depicts the interaction between the phases of the process of analysis defined in this thesis (grey and black) and those from the dependability benchmarking process (white). The main two phases of the analysis make reference to its **definition** and its **application**, and are nec-

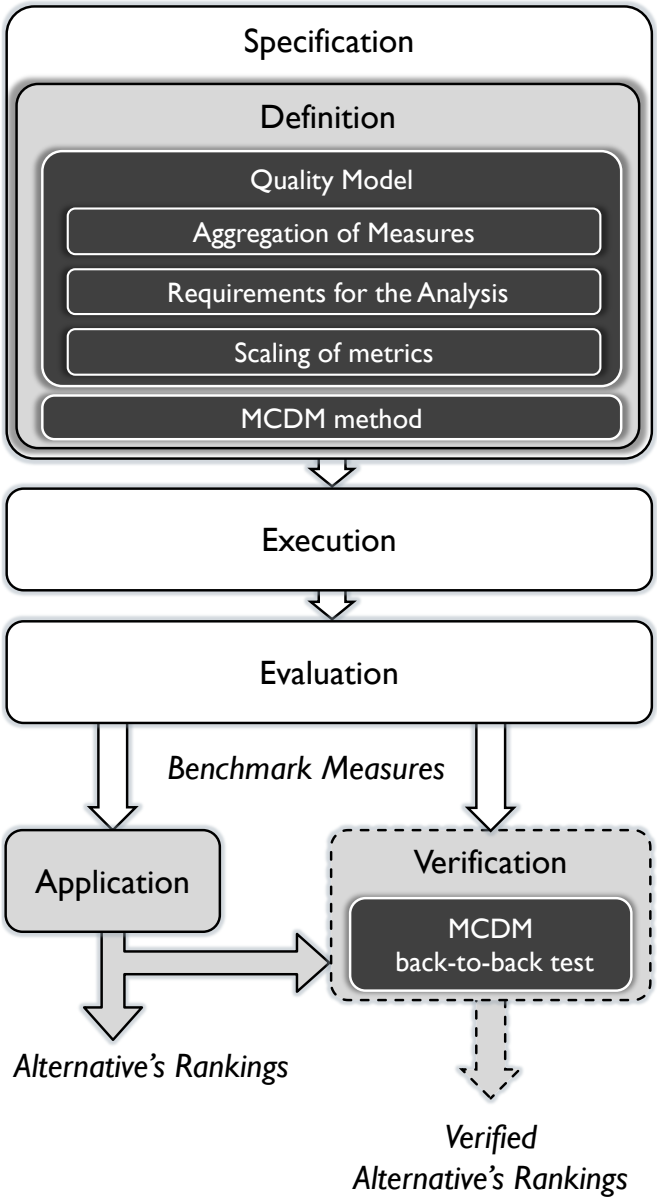

Figure 2.1: Integration of the main phases of the analysis process (soft grey) and the verification phase (dotted border) and the elements that compose them (black), within the dependability benchmarking phases (white)

essary to carry out the analysis and provide the scores to rank the alternatives. The last phase, the **verification** of the analysis, is an additional phase designed to verify that no errors have been introduced in the implementation of the analysis from its definition in the first phase.

The *definition* and *verification* phases of the analysis process, are described in this chapter to provide the reader with a broader view of the work done, which will be useful to understand the contributions presented in next chapters.

However, the *application* of the analysis is not covered in this chapter, as it is better illustrated through the actual application of the methodology to analyze benchmark results from different domains. This is shown in upcoming chapters, where the reader will see examples and detailed descriptions on how the analysis is performed on data from real case studies on dependability benchmarks in different application domains.

To better understand how the work done during this thesis has led to the development of this methodology, a brief introduction of the work presented in the following chapters, and their contribution to the methodology, is provided in the closing section of this chapter.

2.2 Definition phase of the analysis

The **definition** phase of the analysis is the first, and most important of them all. A detailed and clear definition of the analysis is key to guarantee the satisfaction of those benchmarking properties currently missing in the process of analysis.

To specify the procedure to be followed in the analysis, the DM (a role that can be played by the benchmark developer or by another party, like the benchmark user) needs to know the type of measures that will be provided by the benchmark (in which units and scales they will be expressed), and the context of application to determine which measures are more important than others. Since these two aspects of the benchmark are defined during the *specification* phase of a dependability benchmark procedure, the analysis process can be defined in this phase of the dependability benchmark too (see Figure 2.1).

If results are available before the analysis is defined, an evaluator might (unconsciously or not) fine tune the analysis to meet some already expected conclusions. Thus, the early definition of the analysis process provides a more objective interpretation of the results and limits the intrusiveness of the evaluators. The definition of the analysis must also be clear, and every aspect of the analysis has to be detailed during its definition so it can be **repeated and reproduced** by anyone.

As it has been mentioned in the previous chapter, this thesis has studied the feasibility of a specific type of MCDM methods to perform the type of analysis required in dependability benchmarking. These methods provide a mathematical approach that determines how measures must be aggregated in order to characterize the global behavior of the system into a single score, which would enable ranking or comparing different systems to select the one that best suits the evaluator's needs. Nevertheless, there is a set of key elements that must be defined for the analysis to provide meaningful conclusions for the evaluator: **i)** how the measures should be aggregated to provide a global score to characterize the system, **ii)** the relative importance among measures to calculate the global score, and **iii)** how the measures should be homogenized to operate with them during the aggregation process. These elements constitute what, in the context of this thesis, is understood as the *quality model* for the analysis.

2.2.1 Definition of the Quality Model

Inspired in the software quality model proposed by the ISO/IEC 25000 (SQuaRE) standard [62], the quality model (QM) is the formalization of how the benchmark measures must be interpreted and the relationship among them. It is built by the DM, who requires to know in advance the *benchmark context* and the *final measures* of the dependability benchmark. The knowledge on the final measures of the benchmark and their units and scales let the DM define how the measures should be aggregated in the analysis (*aggregation of measures*), and what procedures have to be used to homogenize their values, so they can be aggregated (*scale of metrics*). The benchmark context provides the DM with the necessary information to make the analysis more **representative** for that given context. This information allows her to define the *requirements for the analysis*, where the relative importance among measures is determined, weighting them differently based on their contribution to obtain the global score for the benchmarked system.

Aggregation of measures

When it comes to the analysis of results in dependability benchmarking there might be situations where we must deal with benchmark users with different expectations from the analysis. On one hand, we have *benchmark users* with a more academic profile that might be more eager to have all the possible measures available, so they could perform their own in-depth analysis of the results and promote data sharing among community members [66]. On the other hand, *benchmark users* with a more pragmatical viewpoint, such as those coming from the industry, with more strict time-to-market requirements for their products, might prefer a small set of meaningful and representative scores to characterize, rank and compare benchmarked systems [42].

With the aim of satisfying different kind of profiles during the analysis, this thesis promotes the aggregation of measures following a hierarchical structure, starting from the final measures provided by the benchmark, and ending in the global score that will finally characterize each evaluated alternative. This kind of representation allows the navigation from a fine-grained point of view where all measures are available, to a coarse-grained point of view with fewer (yet representative) values, like it would be the case of the global score. In this way, independently from their viewpoint, benchmark users can interpret results at different levels, and keep track of the origin of the final score, as well as those from intermediate levels.

From here on, to be consistent with the terminology used in the field of operational research, the measures involved in the analysis will be referred as criteria. As the number of criteria grows, so does the number of aggregations that must be defined, which will increase the complexity of the aggregation scheme. Therefore, during its definition, the use of *plain language* must be avoided, as it can be ambiguous, and ease the omission of crucial aspects that can lead to errors during the implementation. Instead, the use of formal methods, like graphical support, is both more intuitive and clear for anyone to understand all the aggregations performed in an analysis.

To provide an example, let us consider the analysis of results from a dependability benchmark to assess the behavior of ad hoc routing protocols ([85]) in the presence of faults. The criteria that will characterize the behavior of the system are the following: The average *throughput* of the network in “kbps”; the average *delay* of sent packets in “ms”; the *energy* consumed during the benchmark expressed in “Joules”; the *availability* of routes between nodes when required, in “percentage”; and the “percentage” of packets whose *integrity* was preserved. Figure 2.2 depicts one of many possible aggregations that could be done by a DM with these five criteria. For this particular case, *throughput* and *delay* are aggregated into a more generic criterion that would characterize the *performance* of the system, and the aggregation of the *availability* and *integrity* measured in the system will characterize its *dependability*. These two upper level criteria are aggregated with the *consumption* criterion into the global score that will represent the overall behavior of the system.

This structure enables the comparison or ranking of a set of alternatives based on their global score, which is one of the main goals of dependability benchmarking. But alternatives can be compared at different levels, *performance* or *dependability*, for example. Actually, global scores can be tracked down to lower levels, so for two alternatives presenting similar scores, the hierarchical structure from the analysis would allow comparing them at a lower level, and perform a decision based on their results in intermediate criteria.

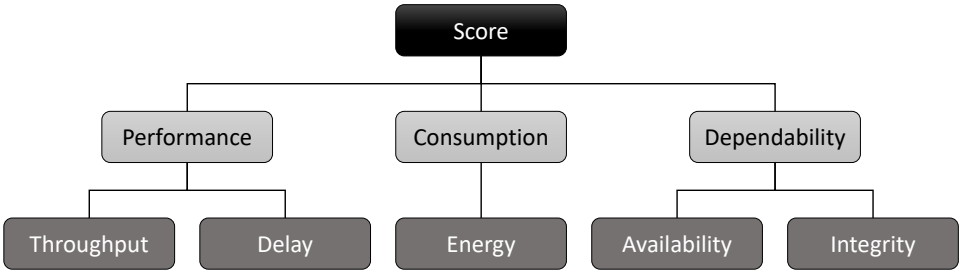

Figure 2.2: Hierarchical representation of the aggregation of criteria for a particular case of ad hoc routing protocols evaluation.

Anyhow, the aggregation of measures is only the first step towards building the QM for the analysis. Once it is defined, the contribution of each criterion to their immediate upper criterion must be quantified by the DM, who has to define these *requirements of the analysis* based on the benchmark context.

Requirements of the analysis

The decision process a person applies to select one alternative among others, independently of the situation, it is always determined by her perception of the context of use. When making a decision among several options, people consider the attributes of each option, and select the option with the best combination of attributes for their needs. That decision process occurs because people actually consider some attributes to be more important than others, hence selecting that option closer to what they consider to be the best solution. However, if someone else is presented with the same problem, and she has a different idea of the best solution, the relative importance among attributes that she considers will be different.

Determining which features of the system under benchmark are more relevant than others will be conditioned by the final purpose of the system under benchmark. In the same way that the *benchmark context* is considered to define representative loads (work-, fault- and change-load) for a benchmark, it also has to be considered to provide a representative analysis process for the benchmark. From the benchmark context, the DM needs to determine which criteria are more important to be satisfied, and so quantify their contribution to the solution (score characterizing the system) accordingly. Since a hierarchical aggregation of criteria is used, the value of all these criteria (except from the bottom-level ones, whose values are provided by the execution of the benchmark) is obtained from the aggregation of the values of their direct sub-criteria.

The contribution of each criterion to its direct upper-criteria is expressed in terms of weights. The weights of all the criteria of an aggregation must add up to 100% if weights are expressed in percentage units, or 1 if they are expressed in a [0,1] scale.

Let us revisit the example from Figure 2.2 and assume the following context for such benchmark: "The nodes of the ad hoc network on which the routing protocols are assessed are static, they are constantly powered, and they are deployed creating several routes between every pair of nodes in the network. The average amount of data exchanged can vary during the day, and there might be peeks of traffic at some point in the day. Most of the data is considered to contain sensitive information". This description of the benchmark context is done in natural language, and its interpretation to define the analysis process might differ between DMs. Here, the use of formal procedures contributes to clarify these interpretations, so any DM can know what decisions have been made in the analysis. Figure 2.3 depicts an example of the requirements that a DM could define for the analysis based on the information about the context. Since nodes are constantly powered, although it is not entirely irrelevant, their *consumption* is not key to make a decision. Due to the sensitivity of the information, assuring the *dependability* of the system is more important than its *performance*. This sensitivity, added to the fact that the deployment is done to guarantee the existence of several routes among every pair of nodes, the *integrity* of the packets stands out over the *availability* of the routes.

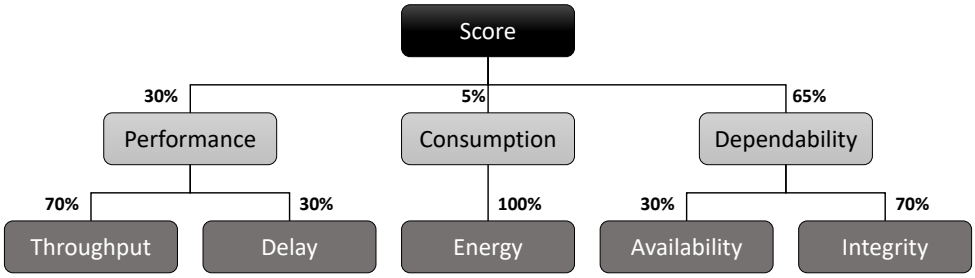

Figure 2.3: Example of weighted criteria that represent the requirements of the analysis for a particular context.

The definition of these requirements is perhaps one of the most difficult parts of the analysis. That is why having detailed and clear information on the context of application is key for the DM to characterize the requirements and make the analysis the most representative possibility for a context. However, the larger the number of criteria to aggregate, the more complicated it becomes to distribute the weights in a meaningful way. This problem is known as *the law of comparative judgement* [106]. It states that while determining the relative importance among two elements is straightforward, peo-

ple find it difficult to provide reasoned weights to quantify the relative importance among a large set of elements. The approached followed in this thesis to deal with this problem is detailed in later chapters.

Providing benchmark users with this information in advance ease the understanding and reasoning behind the analysis, hence giving them the chance to argue either in favor, or against its adequacy for their purpose, and maybe provide alternative requirements. But one thing is clear, considering all criteria equally important leads to performing a context-less analysis, and most of the times it does not represent a real scenario.

Scaling of metrics

The heterogeneity and diversity among the results in dependability benchmarks presents a problem to operate with them. To calculate all the scores on the criteria at different levels, it is necessary to aggregate the values through a given mathematical procedure. However, most mathematical methods require that results are homogenized, and therefore expressed in the same units, in order to operate with them.

There exist a wide number of techniques that can be used to normalized the data in order to work with them. Some MCDM methods have already pre-defined procedures on how to normalize the data from different alternatives whenever required. The authors in [53] identify the four more common normalization procedures used by MCDM methods to normalize the data. These four procedures are presented in Table 2.1, their formula and the scale that determines the range where normalized values will fall is shown alongside them. Here, x_{ij} represents the value of the *i-th* criterion in the *j-th* alternative, while v_{ij} is the result of the normalization procedure for that value. The $\max_i$ and $\min_i$ variables represent the maximum and minimum value, respectively, obtained for the *i-th* criterion considering all the alternatives $(j = 1, 2, \ldots, n)$.

Procedures 1, 3 and 4 preserve the proportion among values after normalization, which means that for the values of two alternatives in a given criterion, let's say the *i-th* criterion, then $x_{ij}/x_{ik} = v_{ij}/v_{ik}$ stands true for every two alternatives. Procedure 2 does not keep this proportion. It defines a maximum and minimum value to establish a range ($[\max_i x_{ij}, \min_i x_{ij}]$ in the formula) where the values are normalized. Depending on the situation, the min and max values of the range can be defined from the values observed in a data set, or determined by the evaluator instead. When not extracted from the data set, values higher than max are normalized to 1, while those below min to 0. While this normalization procedure disperse normalized values along the [0,1] scale (both inclusive), the rest tend to group the values together, although with procedure 3 the normalized values have a tendency to be grouped in the lower part of the scale.

Table 2.1: Normalization Procedures applied in MCDM methods [53]

Procedure 1:	$v_{ij} = \frac{x_{ij}}{\max_i x_{ij}}$	$0 < v_i \leq 1$
Procedure 2:	$v_{ij} = \frac{x_{ij} - \min_i x_{ij}}{\max_i x_{ij} - \min_i x_{ij}}$	$0 \leq v_i \leq 1$
Procedure 3:	$v_{ij} = \frac{x_{ij}}{\sum_{j=1}^{n} x_{ij}}$	$0 < v_i < 1$
Procedure 4:	$v_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}}$	$0 < v_i < 1$

Deciding which technique should be used to normalize the data from the benchmark falls in the hands of the DM, who will be the one deciding which MCDM method should be used for the analysis. However, benchmark criteria can be of two types regarding their meaning, either **benefit** criteria (the higher the value is, the better) or **cost** criteria (the lower the value is, the better). Most of aggregation techniques for MCDM methods require all data to be expressed in terms of benefit criteria for the aggregation to be of any meaning.

Since the procedures from Table 2.1 are defined to normalized data that is only expressed in benefit terms, it is necessary to adapt the normalization techniques to support the data from cost criteria too. *J.J. Dujmovic*, the creator of the MCDM method known as the *Logic Score of Preferences*, presented in [36] a normalization procedure based on *procedure 2* (see Table 2.1), which has been adopted in this thesis. Equation 2.1 and Equation 2.2 show the mathematical formulation to normalize data from benefit and cost criteria, respectively, in a [0, 100] scale.

$$v_{ij} = \begin{cases} 0, & x_{ij} \leq Tmin_i \\ 100\frac{x_{ij} - Tmin_i}{Tmax_i - Tmin_i}, & Tmin_i < x_{ij} < Tmax_i \\ 100, & x_{ij} \geq Tmax_i \end{cases} \tag{2.1}$$

$$v_{ij} = \begin{cases} 100, & x_{ij} \leq Tmin_i \\ 100\frac{Tmax_i - x_{ij}}{Tmax_i - Tmin_i}, & Tmin_i < x_{ij} < Tmax_i \\ 0, & x_{ij} \geq Tmax_i \end{cases} \tag{2.2}$$

Dujmovic considers the maximum and minimum values of the range as *thresholds* that limit the set of acceptable values for a given criteria. Hence, the maximum threshold

($Tmax_i$) indicates, for benefit criteria, the best acceptable value, so any higher value is normalized to the highest value in the normalization scale ([0,1],[0,100],etc). In the other hand, for cost criteria, it represents the worst acceptable value, and any value higher than $Tmax_i$ is normalized to the lowest value in the normalization scale. The minimum threshold, $Tmin_i$, plays the opposite role to $Tmax_i$ for both type of criteria.

Independently from which normalization procedure is chosen, the important thing is that it is made explicit during the **definition** phase of the analysis, so its repeatability and reproducibility is guaranteed. To keep results across works comparable, not only the normalization process must be the same, but the values that characterize such normalization, like min and max values, need to be the same. For example, if the results from a benchmark are normalized using procedure 1 from Table 2.1, others willing to compare their results against those would need to normalize the data using the same value of $max_i x_{ij}$ for every criterion.

As the use of graphical support to make the process explicit contributes to its better understanding, a complete specification of the quality model would involve both, a hierarchical representation of the problem, as depicted in Figure 2.4 (including the thresholds for each criterion), and the definition of the normalization equations (Equation 2.1 and Equation 2.2).

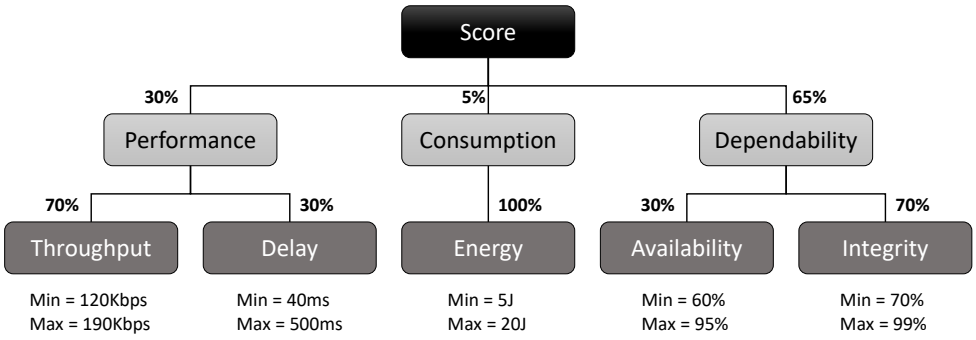

Figure 2.4: Explicit representation of the attributes of the quality model for the analysis.

Once the quality model is complete, there is only one thing left in the definition of the analysis process, to determine which MCDM method will be used.

2.2.2 Multi-criteria decision-making method for the analysis

MCDM methods have been designed to deal with problems involving the comparison and ranking of alternatives based on their performance in a set of criteria. In the context of dependability benchmarks, they can be used to compare and rank benchmarked systems and assist evaluators to determine which one, or set of them, constitutes the most suitable solution to their problem.

This thesis provides a methodology for the analysis process that can be used regardless the MCDM method selected. Given the large amount of available MCDM methods, the feasibility of this methodology is verified by integrating the use of well-known and widely used MCDM methods. The main requirement is that the selected method must be suitable to work with a hierarchical decomposition of the analysis process (as it is proposed for the quality model). For the purpose of illustration, from the suitable methods available, three well-known methods are used in this work to test the feasibility of the proposed approach, and its flexibility to be used with different MCDM methods. These three methods are i) the *Logic Score of Preferences* (LSP), the *Analytic Hierarchy Process* (AHP) and the *Weighted Sum Model* (WSM).

Although they are described in detail in upcoming chapters, in order to provide the reader with basic knowledge about their properties, a brief description for each method is provided in this section.

Weighted Sum Model (WSM)

The WSM [45], based on the *weighted arithmetic mean*, is one of the most commonly used and easiest to understand (and apply) MCDM methods. Its widespread use in IT systems ranges from software selection [64] to risk evaluation of COTS-based systems obsolescence [118].

To perform the aggregation, it is necessary that the data is expressed in the same units and scale, hence data should be normalized if this is not the case. This method does not impose any particular procedure regarding the normalization of the data, so this decision entirely depends on the DM. However, since this method is based on addition, it is necessary that all criteria are expressed in terms of benefit criteria (higher values are better than lower ones). The score for every intermediate-level criterion is computed according to Equation 2.3.

$$S_k' = \sum_{i=1}^{N} w_i \times s_i \tag{2.3}$$

The scores at the bottom-level criteria are obtained directly from the final measures provided as a result of the application of the benchmark. The score of a top criterion in an aggregation is calculated from the scores of the N sub-criteria aggregated to it. The score for each direct sub-criteria, s_i, is multiplied by its weight in that aggregation, w_i. When the process is applied at all levels, the final score of an alternative is calculated and therefore can be used for comparison and ranking purposes.

Analytic Hierarchy Process (AHP)

The popularity of the AHP [91] for organizing and analyzing complex decisions, like selection policies in heterogeneous wireless networks [93], or trust models in Vehicular Ad Hoc Networks [94], is in part originated by its feasibility to compare alternatives based on both quantitative and qualitative criteria.

A numerical scale, depicted in Table 2.2 and known as the *fundamental scale of absolute numbers for pairwise comparison*, is used by DMs to quantify the level of importance among every pair of criteria.

Table 2.2: The fundamental scale of absolute numbers for pairwise comparison

Definition	**Description**	**Intensity***
Equal	A and B are equally important	1
Moderate	A is somewhat more important than B	3
Strong	A is much more important than B	5
Very strong	A is very much more important than B	7
Extreme	A is absolutely more important than B	9

* Intensities of 2, 4, 6 and 8 can be used to express intermediate values. Very close importance values can be represented with 1.1–1.9.

Using this scale, DMs compare criteria two-by-two to identify which criteria are more important to contribute to the upper-level criterion, and to quantify this contribution. These pairwise comparisons among criteria are used to build what is known as a *pairwise comparison matrix*, from which the contribution of each criterion to the aggregation is calculated. This matrix, depicted in Equation 2.4, stores the values defined by the DM from pairwise comparing a set of L elements, where the i^{th} element is represented by M_i, being $i = 1, 2, \ldots, L$. The importance between two elements M_i and M_j is represented as x_{ij}, and the opposite intensity is calculated as $x_{ji} = 1/x_{ij}$, which makes the matrix reciprocal. So, $\forall i, j \in L : x_{ij} \times x_{ji} = 1$.

$$\begin{array}{c} \\ M_1 \\ M_2 \\ \vdots \\ M_L \end{array} \begin{array}{c} \begin{array}{cccc} M_1 & M_2 & \cdots & M_L \end{array} \\ \begin{pmatrix} 1 & x_{12} & \cdots & x_{1L} \\ x_{21} & 1 & \cdots & x_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ x_{L1} & x_{L2} & \cdots & 1 \end{pmatrix} \end{array} \tag{2.4}$$

The corresponding weights for the L criteria are obtained by computing the *priority vector* of this matrix. The priority vector can be computed by either one of the following methods: the *eigenvalue* method [90], or the *row geometric mean* (RGM) method [29]. The difference in the output from the application of these methods is insignificant [31, 59] and therefore both options are acceptable. But since the RGM method requires in general less computational power it is more commonly used.

The procedure followed by the RGM method to compute the priority vector is depicted in Equation 2.5. Its application can be simplified in three successive steps: *i)* compute the geometric mean for each row of the pairwise comparison matrix, *ii)* sum up all computed geometric means, and *iii)* divide each geometric mean by the resulting sum. The result is a priority vector $w^{'} = (w_1, \cdots, w_L)$ containing L weights, where every weight calculated $\forall i \in L : w_i \geq 0$ and $\sum_{i=1}^{L} w_i = 1$.

$$w_i = \frac{\sqrt[1/L]{\prod_{j=1}^{L} x_{ij}}}{\sum_{i=1}^{L} \left(\sqrt[1/L]{\prod_{j=1}^{L} x_{ij}} \right)} \tag{2.5}$$

The interesting thing about the AHP, is that this comparison process, used to determine the weights of criteria for the analysis, is the same that is used to compare the results from different alternatives. Here, the data of each alternative is compared against that from the rest of alternatives for each bottom-level criterion at a time. Hence, instead of normalizing benchmark results through a normalization procedure, for each criterion, the DM uses the scale presented in Table 2.2 to compare, two-by-two, the results from all the alternatives. These comparisons generate a pairwise comparison matrix that, in turn, generates a priority vector that indicates the contribution that each alternative has for that criterion. Therefore, the AHP does not require to normalize the data in advance, since the results from this pairwise comparison are already expressed in the same units and scale. Actually, after performing the pairwise comparison, the data in all criteria is expressed in benefit terms, since the fundamental scale rewards with higher values results that are considered better in a criterion (the higher the importance, the higher the value is).

Once it is done, the scores in higher-level criteria are obtained following the same mathematical procedure that is used in the WSM method, and that is presented in Equation 2.3.

However, comparing the results obtained by the alternatives on each criterion through this procedure, requires for the DM to intervene during the application of the analysis, after the benchmark has been executed, when the results are available. For the particular context of this work, this situation is not desired, as the analysis is exposed to any possible subjective evaluation introduced by the DM. Hence, an *Assisted Pairwise Comparison Approach* (APCA) [78] has been developed in this work to deal with this issue when the AHP is applied. Presented in Chapter 6, the APCA limits the interaction of the DM to the **definition** phase of the analysis, safeguarding the *repeatability and reproducibility* of the analysis and reducing the *intrusiveness* that might affect the conclusions. By doing so, the DM will define the analysis before the results are available and therefore subjective evaluations cannot be made from the final results.

Logic Score of Preferences (LSP)

The LSP method makes use of additional mathematical mechanisms that let DMs fine-tune the preferences of the analysis to make it more representative. The main properties of this method are summarized in this section, although for further clarifications the reader can refer to [36, 100].

The application of the LSP requires the data to be normalized through the procedure depicted in Equations 2.1 and 2.2. In this way it makes sure that the data is expressed in the [0,1] scale, and in terms of benefit (higher values indicate a better behavior). But unlike the WSM and AHP that perform the aggregation of scores through the *weighted arithmetic mean*, the aggregation of scores in the LSP is based on the *weighted power mean*, depicted in Equation 2.6. Its main particularity with respect to the previous methods is that it introduces the use of a *generalized conjunction/disjunction* function, denoted by the *andor* parameter, r in the equation.

$$s_i = \left(\sum_{i=1}^{N} w_i \times v_i^r \right)^{(1/r)} \tag{2.6}$$

To better understand the *conjunction/disjunction* disjunctive in the aggregation, it is helpful to first understand the concept of simultaneity and replaceability among aggregated criteria ([35]). On one hand, operations with properties of *conjunction* indicates that the score of the aggregation will benefit from the *simultaneous* satisfaction of the criteria involved. While not achieving good scores in any criteria involved will have a negative

impact on the aggregated score. On the other hand, when satisfying one or more criteria in the aggregation should have a positive impact in the aggregated score, or in other words, when there is a degree of *replaceability* among the criteria, operations with *disjunction* properties should be used. Table 2.3 depicts the 20 levels of simultaneity and replaceability defined by the author of the LSP, where d reflects the *disjunction degree* among criteria, being $d = 1$ the pure disjunction, and $d = 0$ the pure conjunction.

Table 2.3: Symbols and parameters of the *andor* function

Operation	Symbol	d	r2	r3	r4	r5
DISJUNCTION	D	1.0000	$+\infty$	$+\infty$	$+\infty$	$+\infty$
STRONG QD (+)	D++	0.938	20.630	24.300	27.110	30.090
STRONG QD	D++	0.875	9.521	11.095	12.270	13.235
STRONG QD (-)	D+-	0.813	5.802	6.675	7.316	7.819
MEDIUM QD	DA	0.750	3.929	4.450	4.825	5.111
WEAK QD (+)	D-+	0.688	2.792	3.101	3.318	3.479
WEAK QD	D-+	0.625	2.018	2.187	2.302	2.384
SQUARE MEAN	SQU	0.623	2.000			
WEAK QD (-)	D–	0.563	1.449	1.519	1.565	1.596
ARITHMETIC MEAN	A	0.500	1.000	1.000	1.000	1.000
WEAK QC (-)	C–	0.438	0.619	0.573	0.546	0.526
WEAK QC	C-	0.375	0.261	0.192	0.153	0.129
GEOMETRIC MEAN	GEO	0.333	0.000			
WEAK QC (+)	C-+	0.313	-0.148	-0.208	-0.235	-0.251
MEDIUM QC	CA	0.250	-0.720	-0.732	-0.721	-0.707
HARMONIC MEAN	HAR	0.227	-1.000			
STRONG QC (-)	C+-	0.188	-1.655	-1.550	-1.455	-1.380
STRONG QC	C+	0.125	-3.510	-3.114	-2.823	-2.606
STRONG QC (+)	C++	0.063	-9.060	-7.639	-6.689	-6.013
CONJUNCTION	C	0.0000	$-\infty$	$-\infty$	$-\infty$	$-\infty$

The use of the *andor* parameter allows us to adjust the relationship among aggregated criteria to reward or penalize scores based on their sub-criteria's scores. In fact, this parameter can be individually fine-tuned for the distinct aggregations done in the quality model as the DM sees fit. For example, a $d = 0.5$ *andor* represents the "neutrality function", which is used to define an aggregation function that perfectly balances a mix of conjunction and disjunction properties. Since $r = 1$, this behavior is the same as that from the traditional *weighted arithmetic mean*. The different values of r shown in Table 2.3 ($r2, r3, r4$, etc) indicate the value that r gets for a given value of d when the number of criteria being aggregated is 2, 3, 4, and so on.

Chapter 4 provides some examples of the use of this method to perform more complex aggregations of the criteria. In [34] the author shows an extensive set of examples regarding the complex aggregations of criteria and the combination of values of r to fine-tune the conjunction and disjunction properties of the aggregation.

2.3 Verification of the implemented analysis

Benchmark performers/users willing to follow the proposed procedure have to pay attention to a larger number of details during the analysis than what they are used to, hence the possibility to make mistakes increases. With an increase in the complexity of the analysis procedure, this situation becomes more evident as the output of a dependability benchmark consists of a large number of measures. In a scenario like this one, the amount of potential sources of errors during the implementation of the analysis will endanger the confidence that someone can have regarding the correctness of such implementation.

To deal with this issue, it is possible, and optional, to validate the correctness of the procedure implemented to perform the analysis based on a well known and highly used technique, *back-to-back* testing [114]. The main objective of this approach is to detect, and correct when possible, errors that might have occurred during the implementation of the analysis from its definition. Hence, achieving an error-free analysis would have a positive impact in the confidence that benchmark users would place in the correctness of the conclusions obtained from the analysis.

The quality model constitutes the back-bone of the analysis, it establishes the hierarchical aggregation of criteria, the weights that determine their contribution to the solution, and the scaling procedure for the initial values. Therefore, it is necessary to test that all the elements that conform the quality model have been implemented according to their definition. Then, to perform a back-to-back test of the analysis, it is necessary to implement and execute the analysis process applying a secondary MCDM method. However, as mentioned before, the differences in the mathematical procedures among MCDM methods may lead to different conclusions for the same problem. So, for the conclusions of both analysis to be comparable, the secondary MCDM method must share enough similarities with the primary one to enable the back-to-back test.

The back-to-back test defined consists in performing the analysis of the same benchmark results with both MCDM methods, and use the rankings provided at the different levels in the aggregation hierarchy to identify inconsistencies in the process. The application of this test starts by comparing the rankings obtained by both analyses at the top criteria of their hierarchy (the final scores of the analyses). The presence of inconsistencies between these rankings triggers the search for inconsistencies in the rankings obtained at the direct sub-criteria of the aggregation, again, considering both analyses. To identify the source

of the problem, this search for inconsistencies must be extended to deeper levels, until rankings are consistent among analyses, or until the lowest level of the hierarchy has been reached, where bottom-level values are also compared for inconsistencies.

For example, if alternatives are ranked differently between both analyses at one of the bottom-level criteria, it indicates that a mistake has been made during the normalization of the data, otherwise alternatives would classify the same way. But instead, if no inconsistencies are observed at the bottom-level, but the rankings provided by both analysis differ in the upper criterion, that reflects a problem during the aggregation, probably caused by an error made during the assignment of weights. So, this approach let users to identify and correct errors made during the implementation of the scaling procedures, or the application of weights. However, it is worth to mention that not every inconsistency observed between the rankings of the two analyses will always imply that an error has occurred. Despite the fact that compatible MCDM methods must be used, differences among their mathematical approach may still influence the analysis. Hence, the scores may be quantified differently, which can lead to situations were alternatives with similar results will obtain close values in both rankings, but shift positions between analyses.

An in-depth description of this approach is presented in Chapter 5. The feasibility of this approach is tested in that chapter by performing the analysis of the data from a case study on the evaluation of the robustness to penetrations of an ad hoc network. The results will show how the use of this approach can be useful to detect and correct possible errors that can be made during the implementation of the analysis.

2.4 The methodology in the document

The different aspects of the methodology presented in this chapter represents different milestones that were achieved during the process of this PhD. During this time, the work done has been structured in an article format and submitted to revision by experts in the field. A total of 6 of these articles have been used to compose the core chapters of this document, from Chapter 3 to Chapter 8. Although the integration of this methodology to improve the analysis process in dependability benchmarking is the main argument in all the chapters, the focus of the work varies from one another. Therefore, this section is intended to provide the reader with a better understanding of what aspects of the presented methodology are tackled in the chapters ahead.

A basic rule to do research in any topic, is that in order to solve a problem, it is necessary to be able to answer the following question: "*What is the problem?*". The work described in Chapter 3 focuses the point of attention in the relevance that making the analysis explicit has for a dependability benchmark to be **repeatable** and **reproducible**. Through the analysis of the results obtained from a case study in ad hoc networks ([48]), this

chapter shows how an explicit process of analysis assist evaluators to have a deeper understanding of the results. This contributes to avoid a possible misinterpretation of third party results, thus assuring that comparison of results across works can be done following the same analysis approach, and therefore provide consistent conclusions.

However, in order to actually contribute to benefit dependability benchmarks, it is necessary to consider the analysis of results as an intrinsic part to the dependability benchmark process itself. This makes it necessary to identify the different aspects of the process of analysis that contribute to satisfy the properties that a dependability benchmark must have, as stated in the DBench project [30]. Those aspects of the process of analysis are identified in Chapter 4 and their contribution to the benchmark properties are described and assessed in three different case studies extracted from the literature. The analysis performed in these case studies was either not clear, did not provide a global ranking for the alternatives, or was reduced to compare the alternatives by means of only a single metric. By using these case studies the work described shows the feasibility of this methodology to replicate the analysis performed in those works. At the same time, it can be seen how with this methodology the analysis is made explicit, and it provides means to perform a score-based comparison among the different alternatives (solutions) available.

It can be perceived that this methodology can introduce some complexity to the analysis when compared to methods like the *arithmetic* or the *geometric* mean. This increase in complexity can turn out into a problem if it leads evaluators to make mistakes when implementing the analysis. Under those circumstances, having a way to assess the correctness of their implementations would increase the confidence that evaluators can place in their results . To cope with this problem, the *back-to-back* test approach developed in this work (and introduced in previous section) is described in full detail in Chapter 5. Data from a case study in wireless mesh networks is used to illustrate how different types of mistakes made during the implementation can be tracked down to its origin following this approach, and sometimes correct them.

The viability of this back-to-back test approach relies on the fact that this methodology has been designed to allow a single quality model to be used with different MCDM methods. This way, the MCDM method used to perform the mathematical aggregation of values, does so according to the requirements imposed by the DM for the analysis, which represents the quality model of the analysis. Even though different MCDM methods can be used with this methodology, this work mainly focused on those more popular and common in the literature, like the LSP, the WSM and the AHP. This last one, widely used in many fields of research, requires the intervention of the evaluator during the actual application of the analysis, which contradicts with the property of *non-intrusiveness* that must be achieved by a dependability benchmark. When the evaluator has a direct interaction with the data during the analysis process, her subjective compar-

isons between values will influence the final results of the analysis, and this presents two problems. First, the subjectivity in those operations won't allow other evaluators to apply the same reasoning when analyzing other experiment's results, thus making the analysis not *repeatable* nor *reproducible*. And second, by performing subjective comparisons the evaluator can introduce (willingly or not) bias in the results from the analysis. Hence, Chapter 6 presents the *Assisted Pairwise Comparison Approach* (APCA) developed in this work to remove the interaction of the final benchmark user from the application of the analysis.

Up to this point, the methodology had always been tested in different research domains within the field of dependability benchmarking, yet its application is not limited to dependability benchmarks. Therefore, this methodology was used to compare the results obtained from the assessment of three ad hoc routing protocols on a deployment designed to keep cellphone connectivity accessible to people in post disaster situations. This work, presented in Chapter 7, illustrates the benefits of this methodology in situations where taking the most suitable decision can be key to save people's lives (like a post disaster scenario). With this methodology, the requirements imposed by the benchmark context can be mapped into the analysis, making the process clear and explicit for other evaluators, and providing them with means to perform score-based comparisons among alternatives.

It is true though, that since quality models are built upon the subjective interpretation of the context requirements by the DM, other evaluators may not agree with that process of analysis and apply their own, thus jeopardizing the cross-comparison of results among works. This presents a challenge to extend the application and use of dependability benchmarks, as it will depend on the acceptance it gets from evaluators in the dependable (and industry) community. This challenge is tackled in this methodology by making use of techniques from the field of operational research to build quality models through consensus from the opinions performed by a experts in the field, who act as a set of DM. More concretely, Chapter 8 introduces how this methodology uses the *Aggregation of Individual Judgments* (AIJ) technique to support the definition of quality models from the opinions of multiple experts.

With this brief introduction to what the reader will find in the following chapters, the contribution of each individual piece of work to the development of the methodology should be more evident. The same way, it should provide a bigger picture on how this methodology tackles different problems that jeopardize **the process of analysis and comparison of results in dependability benchmarking for computer systems**.

Chapter 3

Analysis of results in Dependability Benchmarking: Can we do better?

Published at:

- International Workshop on Measurements and Networking

Authors:

- Miquel Martínez[1] - mimarra2@disca.upv.es
- David de Andrés[1] - ddandres@disca.upv.es
- Juan-Carlos Ruiz[1] - jcruizg@disca.upv.es
- Jesús Friginal[2] - jesus.friginal@laas.fr

1. Universitat Politècnica de València, Campus de Vera s/n, 46022, Spain
2. LAAS-CNRS, 7 avenue du Colonel Roche, F-31077 Toulouse, France

Abstract

Dependability benchmarking has become through the years more and more important in the process of systems evaluation. The increasing need for making systems more dependable in the presence of perturbations has contributed to this fact. Nevertheless, even though many studies have focused on different areas related to dependability benchmarking, and some others have focused on the need of providing these benchmarks with good quality measures, there is still a gap in the process of the analysis of results. This paper focuses on providing a first glance at different approaches that may help filling this gap by making explicit the criteria followed in the decision making process.

3.1 Introduction

For many years the evaluation of a system's features made reference to the evaluation of those related to its performance. Nevertheless, the need for providing dependable systems in the presence of perturbations, has lead to a current state of affairs in which many people from both academia and industry evaluate the dependability of systems, in addition to their performance, with comparison and selection purposes. This process, usually known as dependability benchmarking in the research community, has been tackled in many works in the literature where it is applied to different application domains, such as *web servers* [37], *on-line database transactional systems*[112], or *automotive systems*[86], among others.

Most of these works base their benchmarking process on the guidelines established in [30], so as to ensure portable, scalable, and non-intrusive procedures that may lead to repeatable and reproducible experiments. Other works, like [16], focus on dependability measurement to integrate into existing dependability benchmarking processes the common practice followed in metrology. But even though remarkable studies can be found on how to evaluate dependability features in many different systems [67], when it comes to analyse the results obtained in the experiments in order to provide meaningful conclusions, it can be found that evaluators base their conclusions in their own criteria. This presents a problem when different evaluators want to compare their results with the ones presented in another work. This fact has been pointed out in studies like [23] where, among other things, raw data from different experiments and evaluators can be shared, analysed and correlated to obtain good quality measures. However, the purpose of this paper is not focused on data sharing or obtaining quality measures from experimentation, but in pointing out a fact that is present in most dependability benchmarking related works performed so far, and that in our knowledge has not been properly considered yet, the **conclusions reproducibility**.

After analysing many works from the literature, like those presented in [67], it can be observed that the most commonly followed approach consists in presenting the raw measures (computed from the raw measurements/data obtained for each experiment) characterising different system's features, and drawing some conclusions from them. The process of how to compute measures from measurements is usually detailed in depth to show the correctness of such process and enabling other researchers to obtain the same measures. However, as mentioned before, conclusions are usually based on the evaluator's criteria (which is not a bad thing), but the process on how the measures are analysed to provide such conclusions is usually missing, making sometimes hard to understand how the evaluator has come up with them. It is known that in order to compare the results obtained from different experiments, all results must have been obtained following the same process, otherwise comparing them would not provide meaningful conclusions. Thus, a question raises: "*starting with the same results, can we consider useful two different conclusions obtained through different criteria?*" Well, this is not a yes/no answer, it depends. All conclusions extracted from results may be right according to a certain criteria, or wrong according to another, and here is where lies the importance of making explicit the considered criteria in the analysis process.

When reviewing dependability benchmark analyses where the criteria used to obtained the conclusions are missing, external evaluators may disagree with these conclusions, and thus state that the work is not correct. But if the criteria were explicitly defined, external evaluators could understand the reasoning behind those conclusions and thus argue about the analysis process, but not about the work done.

Section 3.2 shows a brief analysis of i) different possible profiles for evaluators, who are the consumers of those conclusions drawn from dependability benchmarking studies, and ii) the different techniques applied that lead to those conclusions. An example that illustrates the benefits of using decision support techniques and the lacks covered by them is presented in Section 3.3, followed in Section 3.4 by a discussion about the feasibility of introducing these methodologies into the common dependability benchmarking process. Finally, the main challenges to be faced are summarised in Section 3.5.

3.2 Background

The number of measures obtained when evaluating a system is usually related to the difficulties found to present the results to end users. For that reason, many benchmarks provide a single score for each system. For instance, when observing the different set of benchmarks provided by the Embedded Microprocessor Benchmark Consortium (EEMBC) [38], all of them get a whole bunch of measures (16 in the case of EEMBC's AutoBench 1.1) but provide a single global measure (Automark for EEMBC's Auto-

Bench 1.1) for a system by calculating a geometric mean with all the given measures. But, when providing a set of measures, it should be taken into account that there are different evaluator profiles that may need to consume these measures. For example, while people from academia may want as many individual measures as possible to exactly determine the effect of certain improvements or new configurations in a system, people from industry, in the other hand, could prefer a single global measure for a straight comparison among competing Commercial Off-The-Shelf (COTS) to be integrated into the system. Likewise, there will probably be some other users requiring more than a single measure, to be able to analyse a system according to different perspectives, but not tens of measures, which make the analysis really hard and often meaningless.

There are different approaches to represent and analyse the multiple measures obtained from evaluation. Although each approach has its own particularities, all of them have to face a common problem: *how to characterise decision criteria within a friendly and usable model*. Choosing a certain kind of representation for measures has important consequences in terms of expressiveness. Simplistic approaches may skew in excess the representation of the model, whereas representations with a high expressiveness can add unnecessary complexity to the model or can be cumbersome in its use for decision making. Therefore it is important to find an equilibrium between representing as much information as possible and maintaining a good degree of usability.

Measures aggregation is a common approach usually applied in the community of dependability benchmarking to ease the comparison among systems. However, it is surprising that so far there is still a lack of unified criteria when addressing the aggregation of measures and their subsequent analysis. Common methods applied by users for aggregation range from simple mathematical operations (e.g., addition, arithmetic mean or geometric mean) to more serious and systematic distribution fitting [27] and custom formulae [1] approaches.

Kiviat or radar diagrams [70] are graphical tools that represent the results of benchmarks in an easy-to-interpret footprint. They can show different measures using only one diagram and, although some training is required, the comparison of different diagrams is fairly simple. The scalability of Kiviat diagrams enables the representation of up to tens of measures. However, managing such a huge amount of information may difficult the interpretation and analysis of results. The problem previously stated is solved in [80] throughout the use of an analytical technique named the *figure of merit* which, imposing certain restrictions to the graph axes, synthesises all the measures into a unique numerical value associated to the footprint shape. However, the problem of this solution, as it happens with most techniques using the mean or the median, is that valuable information could be hidden behind a unique number, and consequently, the comparison between systems could result quite vague [6].

Generally, these techniques focus just on aggregating results and do not provide any insights on how to cope with the interpretation of issuing scores. Nevertheless, there are other techniques that can be used to aggregate the measures while making explicit the decision criteria followed. One of these techniques is the Logic Scoring of Preferences (LSP) [100], a method for combining a large number of criteria into one score. In order to achieve this, an aggregation tree has to be built, where the leaves of this tree are the raw measures. An elementary criterion is defined for each measure, where each criterion has a minimum and a maximum value that define the interval containing the accepted values for each specific measure. The values of the obtained measures are then normalized according to these (minimum and maximum) thresholds. All the measures are aggregated into higher-level features using operators and weights that determine the contribution of each low-level measure to the higher-level one. The final result is a global score that can be used to compare the evaluated system.

Yet another technique that makes explicit the decision criteria is the Analytic Hierarchy Process (AHP) [89]. This technique is widely used in other contexts as a decision making process. As happens in the LSP technique, measures are aggregated using a decision tree, where the leaves are the raw measures and the root is a global score for the system. All the measures are compared two-by-two to determine their contribution to the higher-level criterion. This contribution is obtained by computing the principal right eigenvector of the matrix containing the result of the two-by-two comparison. This process is recursively applied to all levels of the decision tree, thus ending with a global score (priority) for each system that allows their comparison.

As can be seen, exiting aggregation techniques can be classified in those just providing a single score, thus enabling a straightforward comparison of systems, and those based on a hierarchical aggregation of measures, usually in a tree-like form, which enables the navigation from raw measures to a single scores through different levels. Although simple aggregation approaches have been used along the years in the field of dependability benchmarking, it is surprising to note that more complex schemes have not been considered yet. Accordingly, it is necessary to study to what extent they could fulfill the requirements of benchmark evaluators and thus prove their suitability for this domain.

3.3 Proof of concept

In order to show the difference between making explicit or not the decision criteria when analysing dependability benchmark results, this case study makes use of the results obtained in [48], where authors evaluate the behaviour of an ad hoc network in the presence of perturbations. For this example, a small subset of the measures obtained in the work

are used to ease the understanding of the whole process. Nevertheless, studies applying these techniques to a large set of measures can be found in the literature like in [36].

The results in Table 3.1 represent the measures obtained from an ad hoc network in the presence of one of the following attacks: *Replay attack*, *Flooding attack* and *Tampering attack*. Due to the adaptation capabilities of ad hoc networks, the system kept working in the presence of the injected perturbations, but their impact could be observed on the system's performance and dependability degradation. The selected measures for the example are described next:

Availability
: Percentage of time the communication route established between sender and receiver is ready to be used.

Integrity
: Percentage of packets whose content has not been unexpectedly modified.

Throughput
: Average throughput of the network in kilobits per second.

Table 3.1: Measures obtained from the study done in [48]

Measure	Replay attack	Flooding attack	Tampering attack
Availability (%)	75.20	65.00	90.33
Integrity (%)	99.44	98.23	62.90
Throughput (Kbps)	70.90	80.18	96.45

Obtained measures will be analysed by two different evaluators using two different techniques to aggregate the results, and determine which attack impacts the network the most. The first evaluator (**Ev1**) will aggregate the results obtained using a geometric mean (like it is done in the EEMBC), while the second evaluator (**Ev2**) will use the LSP technique described before.

The main purpose of Ev1 is to compare the system's behaviour in the presence of perturbations in an easy way, so the geometric mean suits perfectly for this purpose. The scores obtained by Ev1 after aggregating the measures through Equation 3.1 are shown in Table 3.2.

$$\sqrt[3]{Availability * Integrity * Throughput} \tag{3.1}$$

Ev2 is using the LSP technique, which explicitly defines the reasoning behind the decision process through a mathematical model. The decision criteria followed by Ev2 to aggregate the measures is depicted in Figure 3.1 as an aggregation tree. *Availability* and *Integrity* measures are aggregated into a higher-level feature of the system called *Dependability*, and *Integrity* has been considered of more importance than *Availability* to determine the Dependability of the system. It is to note that this is just taken as an example of aggregation, and it does not mean that these two measures represent the dependability of the system as defined in [13]. Ev2 also considers that to determine a global score for the system, *Dependability* is slightly less important than *Performance*.

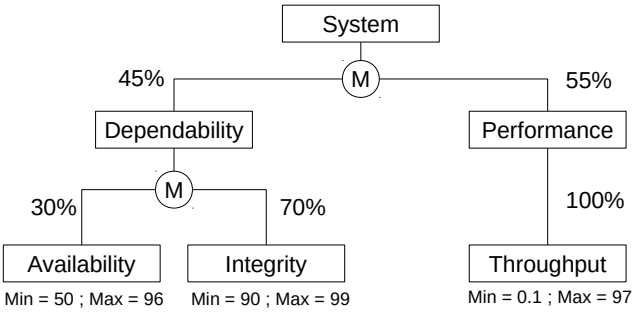

Figure 3.1: Aggregation tree defined by the second evaluator (Ev2) to determine the system score

In the aggregation tree, *Min* and *Max* values represent the threshold values that define the interval of accepted values for each measure. The **M** inside a circle, represents the mean operator, but many different kind of operators can be used depending on the evaluator's requirements. A deeper analysis of these operators is performed in [36], where up to 20 different operators are defined. Table 3.2 shows the scores obtained by Ev1 and Ev2.

Table 3.2: Scores obtained by the first (Ev1) and second evaluators (Ev2)

Evaluator	Replay attack	Flooding attack	Tampering attack
Ev1	80.9359	79.9971	81.8329
Ev2	80.9859	77.2679	55.5521

The different analyses performed by both evaluators lead them to different conclusions. From the results obtained by Ev1, the conclusion is that all the attacks have a similar impact on the system, with the "tampering attack" being slightly more benign, whereas the results obtained by Ev2 show that the "replay attack" has the lowest impact of the three attacks, being the "tampering attack" the most dangerous. As can be seen from

this simple analysis, contradictory results can be obtained from the same set of results just because the interpretation process has not been accurately predefined. Nevertheless, this does not mean that one conclusion is right and the other is wrong. The purpose of the example was to prove that there are other methodologies that can be applied for the analysis of results that provide much more information about the criteria followed by the evaluator when presenting experiment's conclusions. Indeed, Ev2 can also take decisions based on intermediate results issued from the hierarchical aggregation of measures. For instance, the "replay attack" still has the lowest impact on the system from a *Dependability* viewpoint (according to the defined high-level feature). However, when just considering the *Performance* of the system (the other high-level feature), the "tampering attack" is the one providing the best scoring but, as previously described, is the worst case when considering the system as a whole.

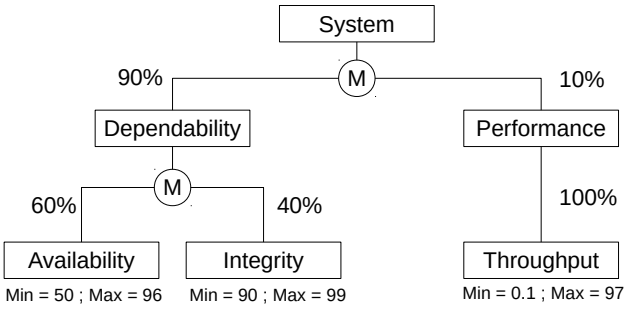

Figure 3.2: Aggregation tree defined by the third evaluator (Ev3)

It is easy to perceive that different aggregation techniques may lead to different conclusions, but there are other problems that may arise from the absence of information about the criteria used. For example, **Ev3** represents another evaluator willing to analyse the data shown in Table 3.1 using the LSP methodology. Ev3 presents the same aggregation tree that Ev2, and also the same thresholds for the different measures but, in this case, Ev3 is considering *Dependability* far more important than *Performance*. Figure 3.2 depicts the aggregation tree with the weights established by Ev3, and Table 3.3 lists the scores obtained after measures aggregation.

Table 3.3: Scores obtained by the third evaluator (Ev3)

Evaluator	Replay attack	Flooding attack	Tampering attack
Ev3	72.8638	58.7766	57.2866

As it can be appreciated from the results shown in Table 3.2 and Table 3.3, both evaluators (Ev2 and Ev3) provide the same classification when ranking the perturbations from low to high impact on the system: i) "Replay attack", ii) "Flooding attack", and iii) "Tampering attack". This example points out the need of making explicit the criteria followed by the evaluator when analysing the results, because whereas Ev3 is considering that *Dependability* features are more relevant to determine the quality of the system in the presence of perturbations, Ev2 considers *Performance* metrics slightly more relevant and, in both case, the same ranking is obtained. Thus, not providing an explicit definition of the decision process may lead readers to misunderstand the reasoning followed to obtain the conclusions, resulting in misleading results when the wrong decision making process will be applied to future experiments performed by that people.

3.4 Discussion

Usually, the criteria used by evaluators is subjective and is determined by the application context of target system. This means that, when evaluating a web server that accesses a database in the presence of attacks, for example, the criteria used to extract conclusions from results obtained should not be the same if that server that manages an industry's private information than if it manages posts in a cooking blog. So, this context is very important and must be taken into consideration when specifying the decision making process to be followed. However, while some of the presented methodologies lack the means to support approach (like Geometric mean or Kiviat diagrams), methodologies like LSP or AHP not only make explicit the criteria used for measures aggregation, but also remove any possible uncertainty in the process, as mathematical models would present less ambiguities than natural language.

A hierarchical representation of the analysis, which enables the navigation from coarse-grain (global score) to fine-grain (raw measures) through medium-grain (intermediate features) viewpoints, opens the doors to evaluators with many different profiles. For example, i) developers may get as many raw measures as desired to have a complete and detailed picture of the system under development, ii) administrators may prefer having a reduced number of aggregated scores characterising different features of the system while tuning its configuration, whereas iii) end users with low expertise may obtain just a single score characterising the quality of the deployed system.

Although the benefits of these approaches seem indubitable, there are a lot of questions still to be solved, like i) how to integrate decision making processes in the common dependability benchmarking process, ii) in case of methodologies being complementary, how can they be combined to make the most of them and ease the decision making process, or iii) in case of methodologies being exclusive, in which scenarios should each

of them be applied. Accordingly, there is still a long way to go before the dependability benchmarking community embraces these practices.

3.5 Conclusions

Along the years, dependability benchmarking has evolved into a mature discipline with applicability in many different areas. Most related works focused on the specification of clear guidelines for the definition and execution of dependability benchmarks, whereas others introduced well formed processes to define good quality measures for the evaluation processes. Nevertheless and although the main goal of these benchmarks is to compare and select among different products or systems those providing the best trade-off between performance and dependability, paradoxically no effort has been devoted yet to provide an accurate and unambiguous decision making process. Common aggregation processes followed to evaluate systems in dependability benchmarking lack rigorousness and vary continuously from one work and evaluator to another. In many cases, the decision criteria applied to analyse the resulting measures is not made explicit, thus making more difficult the fair comparison of results obtained in different experiments and/or by different evaluators.

This work can be considered as a first step forward to pave the way for integrating decision making methodologies into the dependability benchmarking process to enable the **conclusions reproducibility**.

Chapter 4

Multi-Criteria Analysis of Measures in Benchmarking: Dependability Benchmarking as a Case Study

Published at:

- Journal of Systems and Software

Authors:

- Jesús Friginal[1] - jesus.friginal@laas.fr
- Miquel Martínez[2] - mimarra2@disca.upv.es
- David de Andrés[2] - ddandres@disca.upv.es
- Juan-Carlos Ruiz[2] - jcruizg@disca.upv.es

1. LAAS-CNRS, 7 avenue du Colonel Roche, F-31077 Toulouse, France
2. Universitat Politècnica de València, Campus de Vera s/n, 46022, Spain

Abstract

Benchmarks enable the comparison of computer-based systems attending to a variable set of criteria, such as dependability, security, performance, cost and/or power consumption. Despite its difficulty, the multi-criteria analysis of results remains today a subjective process rarely addressed in an explicit way in existing benchmarks. It is thus not surprising that industrial benchmarks only rely on the use of a reduced set of easy-to-understand measures, specially when considering complex systems.This is a way to keep the process of result interpretation straightforward and unambiguous. However, it limits at the same time the richness and depth of the analysis process. This is why the academia prefers to characterize complex systems with a wider set of measures. Marrying the requirements of industry and academia in a single proposal remains a challenge today. This paper addresses this question by reducing the uncertainty of the analysis process using quality (score-based) models. At measure definition time, these models make explicit (i) which are the requirements imposed to each type of measure, that may vary from one context of use to another, and (ii) which is the type, and intensity, of the relation between considered measures. At measure analysis time, they provide a consistent, straightforward and unambiguous method to interpret resulting measures. The methodology and its practical use are illustrated through three different case studies from the dependability benchmarking domain, which usually consider several different criteria including both performance and dependability ones. Although the proposed approach is limited to dependability benchmarks in this document, its usefulness for any type of benchmark seems quite evident attending to the general formulation of the provided solution.

4.1 Introduction

Benchmarks are well-known tools to compare and select distributed systems mainly attending to their performance, cost and power consumption. Standardization bodies, such as the Transaction Processing Performance Council [107], currently propose a set of representative (since widely accepted by the community) benchmarks for distributed systems. In the last decade, some initiatives have addressed the challenging goal of including the evaluation of dependability and security properties in conventional benchmarks. Resulting benchmarks are typically called dependability benchmarks.

Like in conventional benchmarks, controllability and repeatability of experiments and interpretation of results are essential in dependability benchmarks [2, 23, 30]. To date, most of the efforts done in the community around this topic have been oriented towards providing controllability and repeatability of experiments. These efforts can be understood given the need to obtain the same (or at least statistically similar or comparable) experimental measures when the same experimental setup is considered.

However, and without taking importance away from this point, controllability and repeatability also affects other stages of the benchmarking process, such as the analysis of results. The reader should understand that dependability benchmarks introduce the need of performing a more complex analysis of target systems, considering their behavior in the presence of faults and attacks, and characterizing such behavior with a larger set of measures, including dependability and security specific ones. This evidence becomes a challenge when considering the evaluation of complex systems formed by a large and heterogeneous set of sub-systems and components. This is a challenge not only for the amount of measures to consider, but also for their variety of origin and typology.

To date, the analysis of results from dependability benchmarks has been an aspect strongly relying on the human factor. Evaluators subjectively interpret measures following considerations that are usually omitted in the finally generated reports. In consequence, repeating the same analysis of measures and obtaining the same conclusions, even when results are the same, becomes sometimes a complex task.

The underlying problem is that most proposals limit their purpose to the delivery of benchmark measures. Indeed, the consideration of a representative set of measures has been traditionally enough to justify their selection for benchmarking purposes [112]. Then, the analysis of such measures, and consequently the related comparison of alternatives, is typically considered outside the purpose of the specification of most benchmarks, including dependability benchmarks. This can be something acceptable in the context of conventional benchmarks but it is unaffordable in the case of dependability benchmarks, since any aspect leading to a wrong alternative selection may have a negative impact on the safety or security of the system, with the subsequent losses, in the case of critical systems, of reputation, money or lives.

On the one hand, benchmark measures must be contextualized during the analysis process. Without contextualizing their meaning throughout factors such as the environment, the type of system targeted, or the evaluation performer, same results may have different interpretations depending on the evaluation consumer's subjectivity. On the other hand, it must be clearly specified in the analysis process which are the relations considered among measures, and the intensity of such relations. Otherwise, it may be very difficult to guess which have been all the assumptions adopted by someone analyzing a set of benchmark measures. In other words, it may be difficult to verify the conclusions issued from the analysis of a set of benchmark measures.

It is worth mentioning that even if all this effort is done, the analysis and interpretation of results remains an error-prone process requiring a very deep dependability expertise, in the case of dependability benchmarks. This situation increases the *uncertainty* of evaluation analyses and thus negatively affects the credibility of the conclusions obtained. This ambiguous interpretation of concepts is commonly known as *semantic heterogeneity* [5].

This challenge could be addressed through a process of *semantic reconciliation* [5]. Such process involves covering the existing gap between the explicit result of the evaluation, that is, the conclusions distilled from the analysis of measures, and the implicit real intention of evaluators, which concerns the interpretation procedure to obtain such conclusions. This fact increases the sensitivity of analyses, potentially revealing surprising insights about the system under evaluation. This approach is specially useful when there is no obvious optimal (or unanimous) solution due to the large number of criteria that need to be taken into account, or when decisions often require the fulfillment of conflicting objectives (e.g., design or choice of systems maximizing their dependability or performance). It has also the potential for improving the work of system evaluators by leading them to unequivocal and more objective conclusions. Unfortunately, to date, *semantic reconciliation* remains a non-addressed issue in the domain of distributed systems dependability benchmarking.

The main novelty of this paper relies on a double fact. First, providing a multi-criteria analysis methodology to ease the multiple interpretations that the measures issued from dependability benchmarks may have depending on the criteria followed by evaluators. The goal of this methodology is to make explicit the subjective interpretation rules that evaluators typically apply implicitly when determining to what extent measures satisfy evaluation requirements. Doing this in a systematic and repeatable way is essential when different evaluators need to make a fair comparison of their results, so the methodology relies on a mathematical formalism. Second, defining our methodology in such a way it may satisfy the conflicting positions between (i) those evaluation consumers that prefer having all the possible measures as field data for enabling deep result analysis and promote data sharing among community members [66] (e.g., people from academia), and (ii) those adopting a more pragmatical viewpoint that ask for an small set of meaningful and representative scores to characterize, rank and compare evaluated systems [42] (e.g., people from industry). To cope with this goal we rely on the notion of quality model, adopted from ISO/IEC 25000 standards [62], to formulate not only rigorous but also usable and flexible interpretation rules.

Before closing this introduction, it is important to say that the integration of a multi-criteria analysis methodology in very simple benchmarks may be useless, specially where few, or only one, measure or measure type is under consideration. The use of the methodology proposed in this paper makes sense in benchmarking contexts where the analysis process asks for the simultaneous consideration (aggregation and/or comparison) of different measures of different type. The higher the number of measures or the heterogeneity of such measures the higher the usefulness of the proposal. Since this is what happens in dependability benchmarks, the present proposal limits its purpose to this type of benchmarks, and this despite its obvious potential for any other type of benchmarks.

The rest of the paper is structured as follows. Section 4.2 introduces a brief background about dependability benchmarking and multi-criteria analysis. Section 4.3 presents our multi-criteria analysis methodology. Section 4.4 shows the feasibility of our approach through three different case studies and finally. Section 4.5 concludes the paper.

4.2 Background

Computer benchmarks are standard tools that enable the evaluation and comparison of different systems, components, and tools according to specific characteristics [55]. Benchmarks have been widely used to compare the performance of systems (e.g. transactional systems [107] or embedded systems [38]. From a high-level viewpoint, the specification of a conventional benchmark encompasses with the definition of the following components:

- The *system under benchmarking* and the *benchmark target*, which specify the context of use of the system under evaluation and the model of the considered target;
- The *measures* that will be employed to characterize and compare existing alternatives;
- The *execution profile* required to parameterize and exercise both the system under benchmarking and the benchmark target during experimentation. This is typically a *workload*;
- The *experimental procedure* specifying how to run the selected execution profile and how to trace the resulting activity;
- The process to follow in order to transform traces (experimental measurements) into expected benchmark measures.

The main benefit of conventional benchmarks is that, once the set of proposed measures are widely accepted by a community, systems produced by such community can be compared in a quite straightforward and unambiguous way. The key issue here is that most of the considered measures are homogeneous. Indeed, they simply characterize evaluated systems in terms of either their performance or their cost. As a result, comparisons among systems are carried out in a more *representative* way, since based on the use of a set of measures widely accepted by a given community.

Things become however quite different when conventional benchmarks evolved to dependability benchmarks. The seminal work on dependability benchmarking dates from 15 years ago and was produced in the context of the European project *DBench* [30]. Dependability benchmarks characterize the ability of evaluated systems to cope with their

purpose not only in the absence of faults and attacks, as conventional benchmarks do, but also in their presence. The feasibility of the approach and its applicability to different application domains and systems have been shown in [67]. Roughly speaking, dependability benchmarks are specified as conventional benchmarks, but revisiting the concepts of performance profile and experimental procedure as follows:

1. The notion of execution profile is enriched with the specification of a set of accidental faults and attacks, those to which the system must be exposed during experimentation. This set is called the *perturbation-load*.

2. The experimental procedure is reformulated in order to specify not only how considered systems or components must be exercised using the workload, but also how to apply the specified perturbation-load.

Recently, the concept of dependability benchmarking has been also applied in the context of autonomous system, resulting in a new type of benchmark called *resilience benchmark*. In the context of these new benchmarks, benchmarks targets are evaluated, not only in the absence and presence of perturbation-loads, but also in the presence of changes affecting the behavior and/or structure of such targets.

Contrary to conventional benchmarks, the number and heterogeneity of the considered measures is a constant in the various existing dependability benchmarking proposals [67]. Indeed, researchers have proposed, from the very beginning, the use of on-line analytical processing and data warehousing approaches for the analysis and sharing of results from dependability benchmarking experiments [74]. Some other have proposed also the definition of a common repository for sharing the experimental data produced by dependability benchmarks, like the one conducted by the European project AMBER [84]. However, the problem of combining measures in a meaningful and repeatable way was not address by any of these initiatives, although it is of major importance when considering a large number of heterogeneous measures, as in the case of dependability benchmarks.

4.2.1 Comparison of alternatives through aggregation

Measures aggregation is a common approach trying to enable meaningful comparisons among systems that eases the analysis of benchmarked systems or components. However, although these techniques are usually applied in the community of dependability benchmarking, it is surprising that so far there is still a lack of unified criteria when addressing the aggregation of measures and their subsequent analysis. Common methods applied by users for aggregation range from simple mathematical operations (e.g., addition or mean average) to more serious and systematic distribution fitting [27] and custom formulae [1] approaches.

Kiviat or radar diagrams [80] are graphical tools which represent the results of the benchmark in an easy-to-interpret footprint. Kiviat diagrams can show different measures using only one diagram and, although some training is required, the comparison of different diagrams is fairly simple. The scalability of Kiviat diagrams enables the representation of up to tens of measures. However, managing such a huge amount of information may make difficult the interpretation and analysis of results. The problem previously stated is solved in [80] throughout the use of an analytical technique named the *figure of merit* which, imposing certain restrictions to the graph axes, synthesizes all the measures into a unique numerical value associated to the footprint shape. However, the problem of this solution, as it happens with most techniques using the mean or the median, is that valuable information could be hidden behind a unique number, and consequently, the comparison between systems could result quite vague [6].

Other approaches, like the presented in [27], characterize the level of goodness of the measures according to their ability to fit with a particular statistical distribution. Nevertheless, this approach presents three main drawbacks. First, it assumes that a measure follows the same distribution for all the systems, which may be false depending on the context of use. Second, to understand this type of characterization, it is necessary to understand the assumed statistical model, which is not straightforward. Third, the subjectivity of the probability distributions will strongly affect the sensitivity analysis. Finally, it is necessary to handle those situations when there is not enough information to build probability distributions for evaluation data.

Finally, Correia et al. [28] apply the notion of thresholds to map measures into a particular scale for software systems certification. Yet, they assume all the measures have the same importance when it is not always the case.

In sum, previous methodologies lack the ability of aggregating measures into a meaningful way. Generally, these techniques focus on aggregation of results and do not provide any insights on how to cope with the interpretation of the resulting aggregated scores. Accordingly, open questions requiring further research in the domain of dependability benchmarking are (i) how to systematically aggregate such measures to capture in a single or small set of scores the information required to characterize the overall system quality, and (ii) how to ensure the consistency of interpretations issued from the use of such scores with respect to the conclusions obtained from the direct analysis of benchmark measures. Next section is focused on describing how these open questions are coped in this work.

4.2.2 A potential step forward using multi-criteria analysis

The problem of comparing a set of targets according to an heterogeneous set of measures has many similarities with the *multi-criteria decision* problems typically considered in the *operational research* field. So, the use of *multi-criteria decision making* (MCDM) methods to support the analysis of dependability benchmarking measures seems quite promising.

There exist multiple MCDM methods that can be used to address this problem, some of them are widely used in many application domains like business industry, social science, engineering, etc. Among the large number of MCDM methods, some have gained more popularity than others, the *Analytic Hierarchy Process* (AHP) [89] for example, and its use can be found in many works ([73] and [68], for example). Our previous work ([78], [76] and [77]) already presented the feasibility of using MCDM methods to perform the analysis of measures in dependability benchmarking.

The methodology presented in this work will adapt the concepts that apply to MCDM methods with the aim of not only providing mechanisms to better compare different alternatives from benchmarking results, but also to cover the lacks in the analysis that can make dependability benchmarks in particular improve the confidence people from the industry have on them. To that end, next section deeply describes the methodology developed in this work, and its integration in the benchmarking process.

4.3 A multi-criteria analysis methodology to interpret evaluation results

The proposed multi-criteria analysis methodology does not intend to automate the task of benchmarking performers when selecting a proper system; it rather tries to support and guide the comparison of the systems or components fulfilling the system requirements for a particular application, and the selection of the most suitable one.

What makes it interesting for dependability benchmarking with respect to the rest of approaches presented in Section 4.2.1, is its capability to systematize the way to compute the global score of a component not only considering the measures themselves, but also formalizing their interpretation attending to aspects such as the relationship among the measures, and their relative importance within a particular context of use. Accordingly, it is easy to obtain a hierarchical quality model, inspired in the software quality model proposed by the ISO/IEC 25000 (SQuaRE) standard [62], which assists the navigation from the fine-grained measures to the coarse-grained scores without losing the numerical perspective of results. In such a way, one can keep the consistency in the interpretation

and analysis of results independently from the viewpoint (fine or coarse) acquired by the benchmark user.

Figure 4.1 illustrates how the quality model (QM) should be integrated into the dependability benchmarking process, and when it should be applied to provide conclusions from the resultant measures. The definition of the benchmark characteristics in the *experimental set up* lets the evaluator determine the quality model that will later be used to analyze the final measures. The early definition of the analysis process, even before benchmarks are performed, reduces the subjectivity that can be introduced in the analysis process when partial results are being obtained or conclusions are anticipated, which may bias this analysis. This will also ease the cross-comparison among works done from third-party evaluators, as results will be comparable under exactly the same procedure, which may also contribute to the acceptance of dependability benchmarks by the industry.

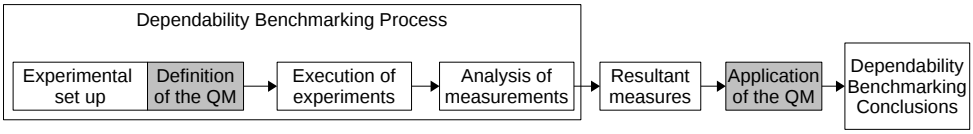

Figure 4.1: Integration of the quality model in the dependability benchmarking process

Defining the quality model according to the requirements of the evaluator (or evaluators) demands the definition of a set of features for the analysis. Upcoming subsections describe these features in detail, identifying their role in the methodology and mapping them to their respective characteristic in the evaluators requirements. The application of the quality model in the analysis process will be later illustrated in Section 4.4 through different case studies.

4.3.1 Benchmark user and target system

The first step is to identify the benchmark targets (in case of more than one alternative), the application context where they operate in and their goal, that obviously depend on the evaluation performer. These aspects are crucial to (i) determine the requirements of the system; and (ii) fix their level of accomplishment.

System requirements can be expressed through the notion of quality model, previously introduced in standards such as [62]. A quality model is a framework to ensure that all the information required by the stakeholder to perform the proper decision-making is taken into account to carry out the analysis of benchmark measures. With respect to this point, the rest of this methodology will introduce the instruments (thresholds, relationships, weights) required to enrich the meaning of measures within the benchmarking process.

4.3.2 Criteria under evaluation

During the *experimental set up*, benchmark performers determine a set of measurable attributes (noted m_1 to m_n) that are representative of the system quality or simply of interest for the evaluation performer. These measures constitute the output of the benchmark, and they are used to compare different benchmark targets and perform the election of the most suitable choice.

In the proposed methodology, the measures defined by the benchmark performer in the first step of the benchmarking process conform the base level criteria of the quality model. These criteria must be understood as the inputs for the quality model that will be used in the analysis process to determine the relative quality of the system according to the defined model. Obviously, the quality and precision of the measures selected in the *experimental set up*, which correspond to the criteria defined in the quality model, will have a high influence on the quality of the conclusions extracted from applying that model in the analysis process. Different works have focused on the selection of attributes in benchmarks to provide good quality measures. Authors in [17] dealt with this problem from a metrology point of view, pointing out the attributes that selected measures must fulfill, so good quality conclusions can be extracted from them. When benchmark performers lack of criteria to determine which measures should be selected, it would be convenient that measures were non-redundant, independent and thoroughly selected attending to their capability to represent quantitative elemental aspects of the system, such as delay, throughput or data availability in a network. This involves that no measure should be derived from other. According to this remark, if we are already taking into account the system's throughput in presence of faults as a measure, considering any other throughput-based measure, such as a ratio between the throughput in absence and presence of faults, would be unfairly providing more importance to throughput than the rest of measures. Despite its importance, and as it has already been considered in other works, the selection of measures is out of the scope of the proposed methodology, that aims at providing mechanisms to improve the comparison of benchmark targets based on the (high quality) resultant measures.

4.3.3 Scales of measures

Given the heterogeneity of the measures considered in dependability benchmarking, it is easy to find different measures using distinct scales and dimensions, e.g., seconds or milliseconds if measuring time, joules if measuring energy, and so on. Obviously, this hinders the analysis and comparison of measures for non-skilled users.

To compare various alternatives, the measures should be brought to the same scale, and normalization methods can be applied to do so. Although normalization methods scale

the values in different ways, they share some common properties. Normalizing by the sum of all the values keeps the proportion between values in the normalized ones. This means, that if a result r_i is the double of r_k, the normalized result v_i will still be the double of v_k. When normalizing by an extreme value (either Max or Min), proportion is also kept, but in both methods, normalized values tend to be grouped together. The use of thresholds, on the other hand, does not tend to group the normalized values but they are distributed along the given range according to their original value.

With the aim of coping with this normalization problem, this methodology propose the use of thresholds within the definition of quality criterion functions $c_i(m_i)$ which specify how to quantitatively evaluate each measure, i.e., they establish an equivalence between the measured value and the system quality requirements within a 0-to-100 quality scale. The result of each criterion function, known as elementary score (or elementary preference), corresponds to s_i. Formally, such elementary preferences s_i can be interpreted as the degree of satisfaction of a measure m_i with respect to the quality requirements specified by the benchmark performer for such measure in the form of a minimum and a maximum threshold (T_{min_i} and T_{max_i} respectively). Since all the measures are scored according to the same normalized scale, resulting elementary preferences are directly comparable. Such equivalence can be mapped to discrete or continuous functions. Equations (4.1) and (4.2) show an example of linear increasing and decreasing functions when measures are the higher the better and the lower the better, respectively. However, these criterion functions can be adapted to satisfy the evaluator's requirements for the normalization of the measures. Examples of how these functions can be adjusted are shown in Section 4.4 through the case studies presented.

$$s_i = c_i(m_i) = \begin{cases} 0, & m_i \leq T_{min_i} \\ 100\frac{m_i - T_{min_i}}{T_{max_i} - T_{min_i}}, & T_{min_i} < m_i < T_{max_i} \\ 100, & m_i \geq T_{max_i} \end{cases} \tag{4.1}$$

$$s_i = c_i(m_i) = \begin{cases} 100, & m_i \leq T_{min_i} \\ 100\frac{T_{max_i} - m_i}{T_{max_i} - T_{min_i}}, & T_{min_i} < m_i < T_{max_i} \\ 0, & m_i \geq T_{max_i} \end{cases} \tag{4.2}$$

The use of minimum and maximum thresholds within criterion functions is necessary to position and compare the value of measures with respect to reference values of the applicative domain, thus easing their interpretation. For example, the interpretation of the measured throughput in a communication system (let us assume 8 Kbps) will be better if the measure is obtained from a Wireless Sensor Network in charge of monitoring temperature (where the optimum value may round 10 Kbps) rather than if it is obtained from

a Wireless Mesh Network to provide Internet access (where even the minimum value allowed for a quality communication, let us assume 500 Kbps, is greater than the value obtained). For each applicative domain, thresholds can be obtained through previous experimentation, the opinion of experts in the domain, or certification and widely-used references. Evaluators or experts in the field should agree on their definition for each measure in a given applicative domain. In this way, comparing the results obtained for different systems is easier, as normalized results are distributed along the range defined by thresholds, instead of being grouped together as happens with other normalization methods. Indeed, the definition of thresholds gives meaning to the values obtained for each measure. Consequently providing the minimum and maximum values that can receive each measure will be very important to determine their preference.

Once measures have been scored, evaluation performers have a founded intuition about the system behavior. In fact, they are able to determine if the individual goal for each particular measure has been accomplished or not. For example, obtaining a score of 75% in one measure could be interpreted as a positive feedback. However, their global preferences about the system requirements are not mapped yet in the result of the evaluation. The idea of the following stage is to aggregate the characteristics of the system according to the evaluation performer's requirements and preferences.

4.3.4 Preferences aggregation

To address the aggregation of scores, this stage of our methodology structures a quality model through a hierarchy of high-level objectives, sub-objectives, etc., where previously computed scores are located at the leaves of the hierarchy. The construction of such hierarchy is relative. First, it is necessary to classify each single score regarding the system characteristic it better fits in. For example, let us assume a transactional system where four measures such as *throughput*, *delay*, *availability* and *reliability* have been considered. In this case, the first level of aggregation could group *throughput* and *delay* within the characteristic of *performance*, and *availability* and *reliability* within the characteristic of *dependability*. This classification of measures can continue grouping similar sub-characteristics into characteristics. Thus, a second level of aggregation would group both *performance* and *dependability* to determine the global quality of the system.

Despite modeling the hierarchical structure of the system, not all the system requirements may have the same importance depending on factors such as the benchmark performer's preferences and the application domain. To cope with this problem, the proposed methodology enables the refinement of the quality model using *weights* to determine the relative importance among requirements for the analysis.

The benchmark performer's requirements that define the quality model should be able to reflect the purpose of the benchmarked system in a given application domain. In some application domains, some measures might be considered of greater importance than others when benchmarking the same system, and thus the quantification of that importance should be implicit in the performer's requirements. Then, the importance that each particular measure has for the analysis is quantified with a weight w_i, where w_i is the weight of the i^{th} particular measure (criterion or resulting sub-characteristic) in a hierarchical level. This measures are weighted according to their relative importance or influence to their direct upper level measure, in such a way that for k measures in a level, $\sum_{i=1}^{k} w_i = 1$. Weights enable to tune the way in which system characteristics contribute to the global quality of the system. Then, consensus between benchmark performers on how measures must be weighted for a given application domain is necessary to contribute to the acceptance of dependability benchmarks in the industry.

Weighting criteria in order to match the benchmark performer's requirements for the context of application of the benchmark is not always easy, and this is particularly important when the model must be accepted by other experts in the field. For that reason, there are different approaches that can be followed to assign these weights. One of them is to rely only in the benchmark performer's criteria and expertise when it comes to determine the relevance of a given criterion against another. Although this might be a good approach, some studies make use of the knowledge and expertise of various experts in the field to define the weights, thus weighting the criteria under evaluation through consensus and agreement. Both approaches are acceptable and used in the literature, however, if someone is seeking for the acceptance of external evaluators, one might think that the second approach would be more likely to be accepted than the first one.

Independently from the approach chosen, weights can be defined directly (using a percentage that determine its importance) or derived using other methods. In [89], the author define a method to quantitatively determine compare the importance of criterion through pairwise comparisons. A 1 to 9 scale known as *Fundamental Scale*, is used to determine the level of importance a criterion has with respect to another. Thus, this scale is used by evaluators to fill matrices that compare criteria at the same level, and the weight for each criterion is derived from performing the eigenvector of that matrix. How the definition of weights should be done entirely depends on the benchmark performer, and thus it is out of the scope of this paper, but more information about this qualitative process of weighting criteria can be found in works like [89] [78] among others.

To illustrate the result of weighting the quality model, let us take into consideration a distributed system within a non-critical solution such as comfort electronic control in cars, probably a rapid response in terms of performance aspects will have more weight than dependability ones (e.g., weighting them 75% and 25% respectively). Conversely, if for example we refer to the Antiblock Brake System (ABS) of the vehicle, evalua-

tion performers may weight dependability above performance assigning weights of 75% and 25% respectively. Fig. 4.2 illustrates this last example. The number above the tree branches indicates the weight assigned in each case.

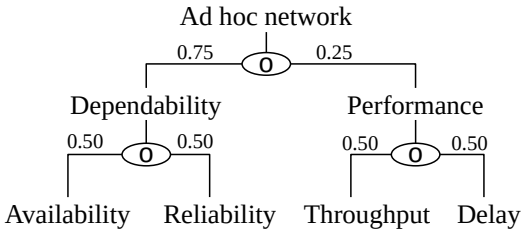

Figure 4.2: Example of weights assignment

Once weights are assigned, it is essential to determine the relation between the elements of the model. For this, different types of operators o may be used to define the conditions under which characteristics are aggregated in Fig. 4.2. The power or generalized mean [22], defined in (4.3), is a generic expression to compute an infinity of aggregation types, considering the notions of scores and weights previously stated. When exponent $r = 1$, this expression is equivalent to traditional arithmetic mean, widely used for aggregation. However, strikingly, the use of different aggregation operators has been rarely considered despite their power to represent, for instance, a punishment in the aggregation result when requirements are not being accomplished or a reward for those requirements that satisfy evaluation criteria. Thanks to (4.3), it is possible to define as many aggregation types as values may take exponent r. Indeed, authors such as Dujmovic propose up to 20 different ones [100]. However, the selection of the proper aggregation operator is a task whose complexity increases as far as more alternatives are considered. Thus, our goal is to define a reduced set of equivalence classes that intuitively represent the different possible levels of aggregation through distinct values of r.

$$S = (\sum_{i=1}^{k} w_i s_i^r)^{\frac{1}{r}} \tag{4.3}$$

To address this challenge, first, it is necessary to introduce the notion of *andness* [119], and how it relates to exponent r. The *andness* of an aggregation operator o, defined in (4.4), is a 1-to-0 coefficient where $andness = 1$ represents that all the system requirements must be satisfied at the same time, and $andness = 0$ involves that just accomplishing any system requirement (regardless which one) is enough.

$$andness(o) = \frac{max(x) - o(x)}{max(x) - min(x)} \tag{4.4}$$

According to [100], $andness = 1$ is associated to $r = -\infty$ whereas $andness = 0$ equates to $r = \infty$. Mathematically, it is quite easy to prove how min is the operator $o(x)$ that makes $andness = 1$, and max is that making $andness = 0$. For the sake of homogeneity, let us denote min with $S+$ to intuitively illustrate the idea that all the system requirements keep a relationship of *strong simultaneity*. Following the analogous reasoning, let max be represented with $R+$ to show the notion that any accomplished system requirement *strongly replaces* the rest (despite they are not satisfied). In the middle, $andness = 0.5$ matches to *arithmetic mean*, which, as previously introduced, is represented with $r = 1$. Let us denote this operator with N to associate its use with the meaning of *neutrality*. Between $andness = 1$ and $andness = 0.5$ there is a gradation of aggregation operators that can be explained as filters that progressively boost the influence of simultaneity against replaceability in system requirements, as far as $andness$ tends to 1. Mathematically, this implies minimising the influence of higher scores while maximizing that of lower ones in the aggregation result. For the sake of simplicity, we have selected $andness = 0.75$ as a representative value of this range. Let us denote this operator of *weak simultaneity* as S. Conversely, the range of operators among $andness = 0.5$ and $andness = 0$ boosts the influence of replaceability with respect to simultaneity as far as $andness$ tends to 0. Similarly, this implies minimizing the influence of lower scores while maximizing that of higher ones. We have selected the aggregation operator with $andness = 0.25$ to represent this equivalence class. Let us denote the *weak replaceability* of this aggregation operator with R. The different values exponent r takes depending on the number of inputs of the aggregation can be found in Table 4.1. For instance, considering the aggregation of 5 different scores with normalized values of 90, 70, 70, 50 and 20, with evenly distributed weights, the final score obtained for operators $R+$, R, N, S, and $S+$ are 90 (max), 72, 60 (arithmetic mean), 48, and 20 (min), respectively.

Table 4.1: Value of exponent r for the operators considered.

Aggregation operators	**2 inputs**	**3 inputs**	**4 inputs**	**5 inputs**
S+ (strong simultaneity)	$+\infty$	$+\infty$	$+\infty$	$+\infty$
S (weak simultaneity)	3.93	4.45	4.83	5.11
N (neutrality)	1	1	1	1
R (weak replaceability)	-0.72	-0.73	-0.72	-0.71
R+ (strong replaceability)	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Previous simple aggregations between scores can be nested to denote those requirements having a special meaning or priority, i.e., a certain degree of mandatoriness or sufficiency for a particular system requirement within the same hierarchical level. For example, Fig. 4.3a illustrates a case where *characteristic A* feedbacks its own simultaneity aggregation (e.g., S), which basically means that satisfying that characteristic is a mandatory condition for the system. Logically, this can be seen as $A \wedge (A \vee B)$, with different degrees of *andness* depending on the selected operators. Thus, not satisfying the requirements of that characteristic would severely penalize the system. Conversely, applying a replaceability operator (e.g., R), would involve defining that characteristic as a sufficient requirement. Likewise, this could be logically expressed as $A \vee (A \wedge B)$, with the selected degrees of *andness*. Fig. 4.3b depicts exactly the same model as Fig. 4.3a but using a simplified notation to ease the use of mandatory and sufficient requirements. Thick branch represents priority requirements in such a way they become mandatory if using S or $S+$ operators, and sufficient if using R and $R+$. To complete this simplification, neutrality operator N and equitable weighs are assumed for the branches omitted. In the rest of the paper the simplified notation will be used.

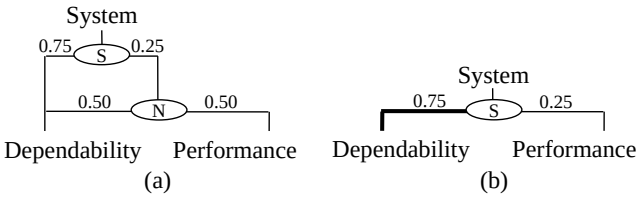

Figure 4.3: Model representing the priority of Characteristic A versus Characteristic B: (4.3a) full model showing how Characteristic A feedbacks its own simultaneity operator (Characteristic A is mandatory), and (4.3b) compact version of that model representing exactly the same hierarchy.

4.3.5 Sensitivity of the quality model

The sensitivity of the quality model is determined by how the sources of uncertainty present in the inputs of the model are translated into uncertainty in the conclusions provided from the application of this quality model.

The aforementioned inputs of the quality model might suffer from a certain degree of uncertainty. For example, errors in the process of measurements (inaccurate measures), a poor understanding of the relevance that each criterion has for the application domain (leading to erroneous weights), or a lack of comprehension of the common behavior of the targeted systems (wrong definition of thresholds). This uncertainty present in the inputs of the model will certainly impact the confidence that benchmark users can place in the conclusions provided as output of the quality model.

Accordingly, the quality model must be analyzed to determine the sensitivity that its output has to the uncertainty in its inputs. This sensitivity analysis can be performed through different methodologies, like those that can be found in [92].

As it was explained in earlier sections, works like [17] have studied the uncertainty from the measurements point of view, setting guidelines to obtain good quality measurements in the system to generate measures with a low uncertainty. Even though studying the uncertainty of the base measures is of prime importance, analyzing the sensitivity of the whole quality model requires a great effort. An extensive analysis on how the combined uncertainty of the inputs of the quality model affect the output conclusions has already been studied in [111] and [25] from the perspective of *multi-criteria decision making* methods.

As the main goal of the paper focused on the definition of a methodology to deal with the analysis of results and comparison of targets in benchmarking, no sensitivity analysis will be done in this work. Nevertheless, this analysis could be very important towards the acceptance of proposed quality models by the industry in different application domains.

Next section presents a set of three different scenarios in the domain of dependability benchmarking that will be used as case studies to illustrate the application of the proposed methodology.

4.4 Case studies

This section shows the feasibility of our multi-criteria analysis methodology along three case studies in the domain of distributed systems, such as web servers, on-line transactional databases and wireless ad hoc networks. As it is possible to apply our methodology at any stage of the analysis (even if measures are already selected, or normalized into scores), as well as to increase the confidence of our study, we apply our methodology from the results delivered by accepted papers in the community. Thus, the information extracted from the papers will be used to elaborate adequate quality models matching author's requirements. The goal is to objectively model the system characteristics to compare the results we are able to obtain through our methodology with those originally delivered by authors. The case studies have been selected in such a way they show the power of our methodology when benchmarking users need to (i) exploit the meaning of measures to properly analyze the system; (ii) rank systems attending to different potentially countered criteria; and (iii) determine the influence that a particular characteristic of the system may have in its behavior. In this way it will be shown the usefulness of the methodology to carry out the analysis of systems following a structured, simple and repeatable way under well-defined evaluation criteria.

4.4.1 Intermediate and global scores to benchmark web servers

In [37], authors perform the comparison of two well-known web servers (Apache and Abyss), running on top of three different operating systems (Windows XP, Windows 2000 and Windows 2003) through the SPECWeb99 benchmark [96]. Thus, authors aim at selecting the best combination of the pair {web server, operation system}. Despite target systems are subjected to 12 different faults encompassing both software and hardware faults, authors finally present only two types of results: those regarding the execution of the system in absence of faults (baseline) and execution in the presence of faults.

Criteria under evaluation

The results of the benchmark are analyzed using 6 measures (3 from performance and 3 from dependability). The set of performance measures is composed of the number of simultaneous connections (con) correctly established (SPECf); the number of operations (op) per second (THRf); and the average time in milliseconds (ms) that the operations requested by the client take to complete (RTMf). With respect to dependability, authors consider autonomy, as a percentage of administrative interventions with respect to the number of faults injected (AUT); accuracy, as a percentage of requests with error with respect to the total amount of requests (ACR); and the percentage of time the system is available to execute the workload from the total (AVL). Table 4.2 collects the results for these measures.

Table 4.2: Measures characterizing the behavior of the pair {web server, operating system} in the presence of faults [37].

System	**AUT** *(%)*	**AVL** *(%)*	**SPECf** *(# con)*	**THRf** *(# op/s)*	**RTMf** *(ms)*	**ACR** *(%)*
Apache-2000	93.98	95.28	13.82	79.24	382.2	97.21
Apache-XP	95.48	97.94	18.07	71.63	359.7	97.60
Apache-2003	96.77	97.62	11.27	79.21	373.1	97.29
Abyss-2000	94.36	96.35	10.32	75.96	363.7	94.78
Abyss-XP	95.97	97.31	13.71	68.22	362.0	94.50
Abyss-2003	96.25	97.53	12.91	66.18	358.7	95.55

Scales of measures

As previously mentioned in Section 4.3.3, thresholds can be determined in different ways. In this case, given the need of authors for ranking systems in the presence of faults, and the lack of field references to determine proper thresholds, an adequate way to get them is using the maximum and minimum values of each measure obtained during the experimentation in the presence of perturbations. This enables a relative comparison between targeted systems in such a way that the maximum value will obtain a score of 100 and the minimum a score of 0. This assignation of scores is suitable when authors are not so interested in the sensibility or meaning of the quantitative measure, since baseline results are not considered, but just in establishing a clear ranking of systems in presence of faults. Thus, we have defined two linear criterion functions $c_i(m_i)$, one increasing for the-higher-the-better measures such as SPECf, THRf, AUT, ACR and AVL; and another decreasing, for RTMf, which is the-lower-the-better, similar to those shown in (4.1) and (4.2) respectively. Maximum and minimum thresholds are shown in Table 4.3.

Table 4.3: Minimum and maximum thresholds for the measures of web servers.

		Thresholds	
Measure	**Function**	**Min**	**Max**
AUT	Increasing	93.98	96.77
AVL	Increasing	95.28	97.94
SPECf	Increasing	10.32	18.07
THRf	Increasing	66.18	79.21
RTMf	Decreasing	362.0	382.2
ACR	Increasing	94.5	97.60

Preferences aggregation

According to authors [37]: *"In this case study we assumed a general-purpose web-server scenario and assigned equal relevance to all six benchmark measures"*. To satisfy such considerations, the quality model has been established following a trade-off solution. In particular, measures have been equally weighted within their category, and neutral operator (N) has been used for the aggregation. The representation of the complete quality model is depicted in Fig. 4.4.

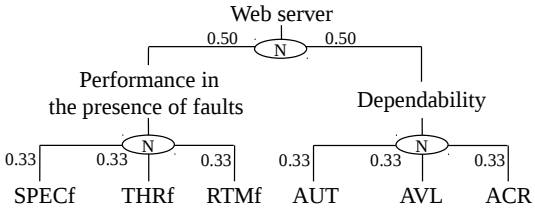

Figure 4.4: Quality model defined for web servers.

Analysis of results

It is worth noting that the results obtained when computing the quality model, shown in Table 4.4, match those obtained by the authors in the paper. When comparing the operating systems for each web-server, *"Windows XP seems to provide the best platform for Apache and Windows 2003 the best for Abyss"*. The comparison of the 6 systems brings up the same conclusions as those given by the authors: *"the combination Apache/XP seems to be the one where the service degradation caused by faults is less noticeable"*. A global score of 81 points quantifies this fact.

Table 4.4: 0-to-100 normalized results (scores) after applying the quality model shown in Fig. 4.4.

System		AUT	AVL	SPECf	THRf	RTMf	ACR	Performance	Dependability	Global score
Apache	2000	0	0	45	**100**	0	87	48	29	38
	XP	54	**100**	**100**	42	96	**100**	**78**	84	**81**
	2003	**100**	88	12	**100**	39	90	50	**92**	71
Abyss	2000	14	40	0	75	79	9	51	21	36
	XP	71	76	44	16	86	0	48	49	48
	2003	81	85	33	0	**100**	34	44	66	55

Apart from that, it is remarkable that scores at leaves are consistent with those delivered at intermediate ones (performance and dependability scores) and the root (global score). As seen, it is possible to navigate from fine-grained to coarse-grained scores through intermediate ones. Indeed, it is possible to discover sensitive information that is not provided in the original paper. Attending to intermediate criteria, it is possible to observe that the pairs {Apache, XP}, with 78 points, and {Apache, 2003}, with 92 points, are the best candidates from a performance and dependability viewpoint respectively. As observed, the use of quality models can be useful to improve the exploitability of measures in the analysis of results.

4.4.2 Managing multiple criteria for comparing OLTP systems

In [112] the authors propose a dependability benchmark for On-Line Transaction Processing (OLTP) systems. Thus, ten targets (A to J) are defined based on the combinations of (i) two different versions (DB 1, DB 2) of a leading commercial Data Base Management System (DBMS), (ii) two DBMS configurations (Conf A, Conf B), (iii) three operating systems (Windows 2000, Windows XP, SuSE Linux 7.3) and (iv) two different hardware platforms (HW 1, HW 2).

Criteria under evaluation

From the benchmarking process, the authors obtain measures based on three different criteria: baseline performance, performance in the presence of faults, and dependability. Such measures, typically used in the TPC-C [107] benchmark, are the *number of transactions (trans) per minute (m)* and *price ($) per transaction*. When these measures are obtained in absence of faults (baseline performance), they are labeled as tpmC and $/tpmC respectively, but when obtained in the presence of faults (performance in the presence of faults) they are labeled as Tf and $/Tf. Dependability measures make reference to the percentage of time the server is available (AvtS), and the percentage of time the client is available (AvtC). Table 4.5 shows the original values of the measures provided in the paper.

Table 4.5: Original measures extracted from [112] characterizing the 4-tuple {operating system, DBMS, configuration, hardware platform}.

System	**tpmC** *(#trans/m)*	**$/tpmC** *($/#trans)*	**Tf** *(#trans/m)*	**$/Tf** *($/#trans)*	**AvtS** *(%)*	**AvtC** *(%)*
A: {Win 2000, DB 1, Conf A, HW 1}	2244	12	1525	17.7	86.1	75.4
B: {Win 2000, DB 2, Conf A, HW 1}	2493	11.6	1818	16	87.2	79.5
C: {Win XP, DB 1, Conf A, HW 1}	2270	11.9	1667	16.2	88	79.4
D: {Win XP, DB 2, Conf A, HW 1}	2502	11.6	1764	16.4	88.6	79.5
E: {Win 2000, DB 1, Conf B, HW 1}	1411	19.1	896	30.1	74.2	68.7
F: {Win 2000, DB 2, Conf B, HW 1}	1529	19	969	29.9	76.6	69.7
G: {SuSE 7.3, DB 1, Conf A, HW 1}	1961	12.7	1406	17.8	86.3	77
H: {SuSE 7.3, DB 2, Conf A, HW 1}	1958	13.8	1400	19.3	93.5	83.9
I: {Win 2000, DB 1, Conf A, HW 2}	3655	7.7	2784	10.1	89.4	79.5
J: {Win 2000, DB 2, Conf A, HW 2}	4394	6.8	3043	9.9	88	80.9

Scales of measures

Given the absence of clear or explicit arguments of authors to carry out the comparison of systems in this case study, let us perform the selection of thresholds positioning the results of their evaluation with respect to referenced values obtained in the community [54] in the last years. This choice pursues a double goal. First, not only to compare target systems among one another in a local way, but also to provide a useful feedback about their behavior when adopting a wider perspective and comparing them with other systems using TPC-C benchmarks, even when they are not subjected to faults. Second, showing the capability of our methodology to incorporate multiple ways to select scales of measurement. Hence, for the definition of thresholds, we have taken into account the results delivered in [54] for the year 2000, when the hardware platforms considered in this case study appeared. Table 4.6 shows the upper (maximum threshold) and lower (minimum threshold) values of the trend for TPC-C in the intersection with that year. It must be noted that tpmC and Tf, on the one hand, and $/tpmC and $/Tf, on the other, represent the same measures but in absence and presence of faults, respectively. This is why the same thresholds are defined for both measures.

Table 4.6: Thresholds determined for the different measures of OLTP systems.

		Thresholds	
Measure	**Function**	**Min**	**Max**
tpmC	Increasing	1400	4800
$/tpmC	Decreasing	1	20
Tf	Increasing	1400	4800
$/Tf	Decreasing	1	20
AvtS	Increasing	74	100
AVtC	Increasing	70	100

Preferences aggregation

The authors classify the ten systems attending, each time, to a different criterion (baseline performance, performance in the presence of faults and dependability). Despite this situation may require the generation of three different quality models, one per criterion considered, it is also possible to generate just one quality model that can be parameterized in such a way that the different cases are represented at the same time. Let us take into account this last alternative to show the expressiveness power of our approach.

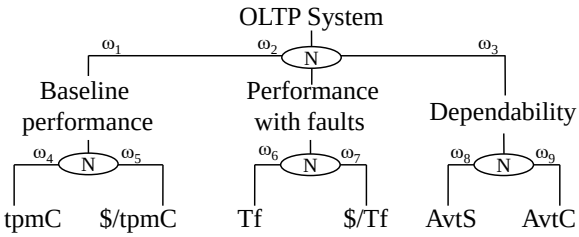

Figure 4.5: Parameterized quality model gathering all the single criterion stated by authors and the proposed trade-off between all measures.

Each branch of the quality model defined in Fig. 4.5 has been assigned a given weight, whose value can be modified as shown in Table 4.7 to model the three different criteria defined by authors. Weights for tpmC, Tf, $/tpmC and $/Tf scores have been properly parameterized, as the last two are not considered by authors in the definition of the classifications. Likewise, being the availability of the server more critical than the exhibited by clients, as explicitly commented by authors, weights have been accordingly adapted. Finally, the authors also propose the generation of a trade-off ranking to reach a consensus between the three criteria previously tackled. Unfortunately, despite they let the reader know that it is based on the previous rankings, they do not structure a clear reasoning on how this classification is achieved. Given the role of our methodology to cover potential ambiguities and lacks of thoroughness, it would be possible to define alternative weights to adequately address the trade-off ranking concerned.

Table 4.7: Weights for the parameterized quality model shown in Fig. 4.5.

Characteristics	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
Baseline performance	1	0	0	1	0	0	0	0	0
Performance with faults	0	1	0	0	0	1	0	0	0
Dependability	0	0	1	0	0	0	0	0.70	0.30
Trade-off	0.33	0.33	0.33	0.5	0.5	0.5	0.5	0.70	0.30

Analysis of results

Table 4.8 shows the intermediate and global scores for each system after computing the trade-off quality model previously proposed. Table 4.9 collects the different rankings to ease the comparison between systems.

Table 4.8: 0-to-100 normalized results (scores) after applying the trade-off weights from Table 4.7 to the quality model shown in Fig. 4.5.

System	Baseline performance	Performance with faults	Dependability	Trade-off
A	33	7	37	26
B	38	16	45	33
C	34	13	47	32
D	39	14	49	34
E	2	0	1	1
F	4	0	7	3
G	27	5	39	24
H	24	1	67	31
I	65	46	51	53
J	78	50	49	58

From the intermediate scores that belong to the different criteria, it can be appreciated that the single criterion rankings match those defined by the authors. Nevertheless, the ranking established according to the trade-off criterion presents a similar, but not equal order. While in the paper the trade-off ranking is "*I, J, D, B, C, H, G, A, F and E*", with the methodology proposed systems "*I, J*" and "*G, A*" swap their positions. The problem, in consequence, is not so the analysis done by the authors, probably correct, but the difficulty to exactly reproduce it again with the tools they provide. This result shows the need for establishing clear and explicit rules when addressing the analysis of benchmarked systems. As observed, the use of quality models can be useful not only to easily rank different systems despite applying different criteria, but also to unequivocally repeat this ranking when needed.

Table 4.9: Original rankings carried out in [112] against those obtained from applying quality models.

		Ranking of systems									
		1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
Baseline performance	Original	J	I	D	B	C	A	G	H	F	E
	Quality model	J	I	D	B	C	A	G	H	F	E
Performance in the presence of faults	Original	J	I	B	D	C	A	G	H	F	E
	Quality model	J	I	B	D	C	A	G	H	F	E
Dependability	Original	H	I	D	J	C	B	G	A	F	E
	Quality model	H	I	D	J	C	B	G	A	F	E
Trade-off	Original	**I**	**J**	D	B	C	H	**G**	**A**	F	E
	Quality model	**J**	**I**	D	B	C	H	**A**	**G**	F	E

4.4.3 Evaluating perturbations on ad hoc networks

This case study aims to show the feasibility of this methodology to determine the impact that each single perturbation has over a system when considering its injection separately from the rest of perturbations compounding the faultload. In [48], the authors perform the evaluation of two different and representative types of ad hoc networks, a static Wireless Sensor Network (WSN) where 6 real nodes execute AODV routing protocol (Network A) and a Mobile Ad Hoc Network (MANET) where 6 real mobile nodes run OLSR routing protocol (Network B), when subjected to perturbations. Such set of perturbations is formed by accidental faults like *signal attenuation* and *ambient noise*; and attacks such as *flooding attack*, *replay attack* and *tampering attack*.

The networks studied on this paper are mapped into a specific context of use, representing each one different situations of the real world. The specifications of each network are represented in Table 4.10.

Table 4.10: Experimental configuration of Network A and Network B presented in [48].

Network	**RP**	**Speed**	**Area**	**Range**	**Workload**
A	AODV	6 nodes: 0 m/s	30 x 50 m	20 m	Text data (500 bps)
B	OLSR	6 nodes: [0-3] m/s	300 x 150 m	125 m	VoIP traffic (100 Kbps)

RP: Routing Protocol

Criteria under evaluation

In the paper, the authors evaluate the impact of each perturbation in the network considering two performance measures: the applicative throughput (or Goodput), and the increment of delay (or Jitter); and two measures of dependability: the percentage of packets correctly delivered (or Integrity), and the percentage of time the network is ready to be used (or Availability). Table 4.11 illustrates the values measured by the authors for each considered perturbation in Network A and Network B.

Table 4.11: Measures obtained from the case study of ad hoc networks.

			Perturbations				
	Measure	**Golden run**	**Signal attenuation**	**Ambient noise**	**Replay attack**	**Flooding attack**	**Tampering attack**
Netw. A	*Availability (%)*	92.94	73.98	88.74	93.89	51.22	90.12
	Integrity (%)	99.03	97.53	92.12	98.54	97.56	8.01
	Goodput (Kbps)	0.19	0.17	0.18	0.19	0.10	0.19
	Jitter (ms)	319.89	353.45	332.66	300.78	721.66	312.44
Netw. B	*Availability (%)*	95.14	73.9	87.00	75.20	65.00	90.33
	Integrity (%)	98.34	98.73	92.26	99.44	98.23	62.90
	Goodput (Kbps)	96.45	85.19	90.56	70.90	80.18	96.45
	Jitter (ms)	199.98	210.23	211.11	220.88	230.55	195.00

Scales of measure

This case study has an interesting detail that can not be found in the previous case studies. Unlike the others, the authors establish a discrete three level criteria (Low, Medium or High) to evaluate the impact of perturbations on the measures: *"In this way, the impact is considered low, medium or high if the measure is degraded underneath 5%, over 5% or over 10% respectively, according to the golden run results"*. Accordingly, (4.5) and (4.6) define a discrete three-level criterion function for the-higher-the-better measures (availability, integrity and goodput), and the-lower-the-better measure (jitter), respectively. In these equations, $B(m_i)$ refers to the baseline computed value for measure m_i.

$$s_i = c_i(m_i) = \begin{cases} 0, & m_i \leq 0.90 \cdot B(m_i) \\ 50, & 0.90 \cdot B(m_i) < m_i < 0.95 \cdot B(m_i) \\ 100, & m_i \geq 0.95 \cdot B(m_i) \end{cases} \tag{4.5}$$

$$s_i = c_i(m_i) = \begin{cases} 100, & m_i \leq 1.05 \cdot B(m_i) \\ 50, & 1.05 \cdot B(m_i) < m_i < 1.10 \cdot B(m_i) \\ 0, & m_i \geq 1.10 \cdot B(m_i) \end{cases} \tag{4.6}$$

Preferences aggregation

After identifying the three different levels quantifying the impact of perturbation on the obtained measures, authors do not detail how to determine the impact of the perturbation on the whole system. Instead, they perform a qualitative analysis (also based on three discrete levels) with no clear rules about how it was perform. Accordingly, as no special requirements for the scores aggregation are defined, equitable weights and neutral aggregations have been considered for all the branches of the proposed quality model shown in Fig. 4.6.

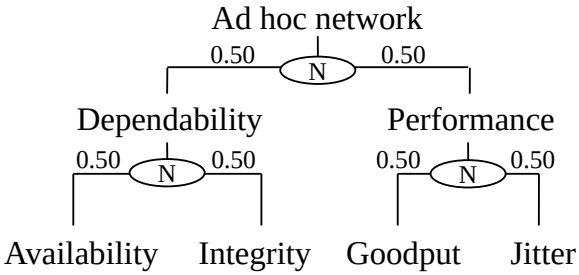

Figure 4.6: Quality model to determine the impact of each perturbation on the considered ad hoc network.

Analysis of results

The global scores obtained for each of the networks are listed in Table 4.12. As previously stated, authors make a qualitative analysis of the impact of each perturbation on each measure to determine the actual impact of the perturbation on the whole system (Low, Medium, High). Since there is no explicit information about how this analysis is performed, we propose to determine the impact level according to the global score obtained for each perturbation. As measures are normalized according to their deviation with respect to the baseline, final scores between 100 and 70 indicate that the perturbation is barely affecting the system (low impact level), scores between 69 and 40 show a medium impact level, and scores between 39 and 0 reflect a high impact.

Table 4.12: Characterization of the impact level according to the scores for Network A and Network B.

	Perturbation	Score	Quality Model Impact level	Original Impact level
Network A	Signal Attenuation	25.0	High	High
	Flooding attack	25.0	High	High
	Ambient noise	75.0	Low	Low
	Replay attack	100.0	Low	Low
	Tampering attack	75.0	Low	Low
Network B	Signal Attenuation	37.5	High	High
	Flooding attack	25.0	High	High
	Ambient noise	50.0	Medium	Medium
	Replay attack	25.0	High	High
	Tampering attack	62.5	**Medium**	**Low**

The resulting classification for perturbations affecting both networks matches that obtained in the original paper, but for the *tampering attack* on Network B, which is now classified as having a Medium instead of Low impact. This divergence obviously derives from the vague description of the characterization performed on the original paper. As in Section 4.4.2, this shows the necessity of precisely defining the criteria and procedure followed during the results analysis. Otherwise, the same results could be interpreted in a completely different way, preventing this process from being repeatable.

In addition to the analysis performed in the original work, and to show the potential of the proposed approach, it could be possible to define a new quality model to help evaluators when deploying a new routing protocol in the network, tuning routing protocol parameters, or introducing new fault tolerance mechanisms, for instance. This model could take into account the information extracted from this case study, so those perturbations presenting a high impact on the system could be aggregated with equal weight under *critical* perturbations category, and those with a lower impact could be grouped under the *non-critical* perturbations category. The severity of critical perturbations could be remarked by punishing those critical scores with a low value. So, a mandatoriness relationship with the simultaneity operator S, could be used to illustrate this purpose. Medium and low impact perturbations could present different weights, like 0.75 and 0.25 respectively, to reflect their different importance. Fig. 4.7 and 4.8 show the resulting quality models for Network A and Network B respectively.

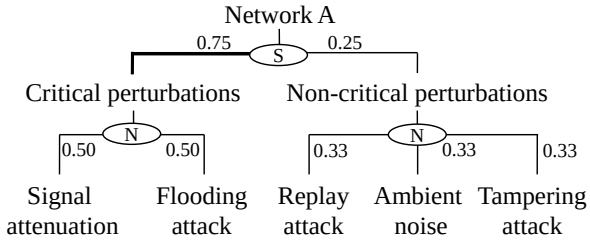

Figure 4.7: Aggregation of perturbations for Network A (WSN).

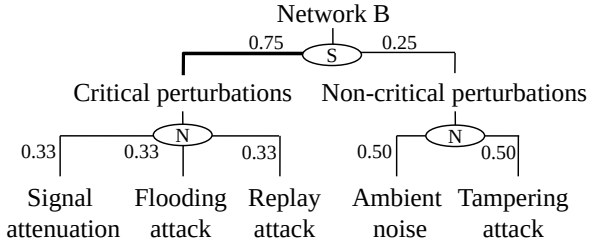

Figure 4.8: Aggregation of perturbations for Network B (MANET).

4.5 Conclusion

In this paper, we have presented a methodology to make straightforward, consistent and objective the analysis of dependability benchmarking measures, a big challenge in todays distributed systems. Our methodology addresses how to adequately select and gather the types of measures to represent the system quality. Since there are distinct ways to do it, our methodology enables the generation of multiple representations (or quality-scores-based models) from the same system when different criteria are applied by evaluators. Among their benefits, the scores obtained from our methodology are repeatable simply following the explicit criteria defined in each quality model, which eases the comprehension of evaluation assumptions, thus assisting the benchmark user to minimize mistakes during the results interpretation. Indeed, the model provided becomes not only a way to express which measures are under consideration, but also a mean to drive their analysis in a more objective and systematic way. Objectiveness is important to minimize the provision of biased conclusions, while the systematization of the approach enables the provision of tools to assist users in the consideration of a big number of targets, faults and measures during experimentation.

Furthermore, our methodology results a very useful approach to overcome the problem of measures scalability and gets a more quantitative vision of the system despite the

multiple aggregation of scores. Nevertheless, regarding previous results, the application of this technique requires the adequate definition of the quality thresholds (X_{min} and X_{max}) for each criterion functions, the weight (w_i) assigned to each score within the same hierarchical level, and the operator type (o_i) in charge of the scores aggregation. All these aspects highly depend on the applicative context the system is conceived to be deployed in. Despite the selection of these parameters may result subjective, our methodology forces the benchmark performer to make them explicit, which eases the transparency and comparison between systems. This is an advantage with respect to traditional benchmarking, where the criteria considered usually remain subjective and hidden to the benchmark report consumer.

The application of our methodology in the case studies presented in the paper begin from a stage of the evaluation where measures are already available, which is very often when authors compare their results. However, conversely to other measures-aggregation techniques, our methodology could play an active role during the benchmark definition, being applied from the very beginning, i.e., before benchmark experiments are carried out. Considering this point is a first step towards improving the characterization of the wide amount of applicative domains in distributed systems. We argue that this type of approaches can be useful not only to quantify the impact of faults with respect to the actual application context (where components and systems are planned to be deployed), but for the comparison and selection of those targets which best fit the system requirements.

In the future work, we ambition to provide evaluators different templates with precomputed parameters that they could customize for their particular deployments to semi-automate the application of this methodology for the quantitative benchmarking of different types of distributed systems.

Chapter 5

Gaining confidence on dependability benchmarks' conclusions through "back-to-back" testing

Published at:

- Tenth European Dependable Computing Conference

Authors:

- Miquel Martínez - mimarra2@disca.upv.es
- David de Andrés - ddandres@disca.upv.es
- Juan-Carlos Ruiz - jcruizg@disca.upv.es

1. Universitat Politècnica de València, Campus de Vera s/n, 46022, Spain

Abstract

The main goal of any benchmark is to guide decisions through system ranking, but surprisingly little research has been focused so far on providing means to gain confidence on the analysis carried out with benchmark results. The inclusion of a back-to-back testing approach in the benchmark analysis process to compare conclusions and gain confidence on the final adopted choices seems convenient to cope with this challenge. The proposal is to look for the coherence of rankings issued from the application of independent multiple-criteria decision making (MCDM) techniques on results. Although any MCDM method can be potentially used, this paper reports our experience using the Logic Score of Preferences (LSP) and the Analytic Hierarchy Process (AHP). Discrepancies in provided rankings invalidate conclusions and must be tracked to discover incoherences and correct the related analysis errors. Once rankings are coherent, the underlying analysis also does, thus increasing our confidence on supplied conclusions.

5.1 Introduction

Since the seminal research carried out during the DBench European project more than 10 years ago [67], lots of efforts have been done in dependability benchmarking resulting in the current availability of a wide variety of dependability, security and resilience benchmarks. The similarities among existing proposals is not surprising, since most of them rely on the DBench experimental framework, which is adapted and extended in each proposal attending to the variety of constraints imposed by each particular system, application domain and/or context of use.

Despite the interest for comparing different component and system implementations, configurations and parametrisations, dependability benchmarking has attained so far a limited industrial adoption. Discussing the root causes of this situation falls beyond the scope of this paper but what seems quite clear is that, in some cases, the requirements imposed to dependability benchmarks by the academia are different from those expected by the industry. Approaches, like the SPEC Research IDS Benchmarking Working Group [104], aims at mitigating that problem by fostering innovative research through exchange of ideas and experiences between academia and industry, although there exists a long way to go.

The vision of industrials of what is a dependability benchmark is usually quite pragmatic; they consider such type of benchmarks as tools to support, or automate to some extent, the process of selecting the most suitable components for the particular type of systems they produce. As a result, they ask for the provision of a limited number of results (if possible one) in order to accelerate and simplify the final selection/decision process un-

derlying any benchmarking effort. On the other hand, researchers prefer to provide the so-called *necessary* (sometimes large) number of measures to establish a precise and well-reasoned ranking among all benchmarked targets. It must be noted that this approach is not a problem by itself. The problem is that different rankings and conclusions can be issued from the analysis of the very same set of benchmarking measures. One of the aspects leading to that situation is the lack of any explicit representation of the analysis procedure followed to issue conclusions, which limits in practice the repeatability of such procedure. This situation should not be a surprise for the reader since analysing benchmarks results refers to a well-known and subjective multi-attribute analysis process [69]. As a result, and despite the pertinence and correctness of conclusions, the analysis performed must be always studied attending to the particular subjective (judgmental) analysis criteria used by the decision maker.

The use of multiple-criteria decision-making (MCDM) techniques provides means to explicitly represent the analysis process followed when interpreting (benchmarking) results under the form of a multi-attribute decision model, also called quality model. Multicriteria decision problems may have different goals that are very close to those pursuit when analysing dependability benchmarks results [69]: i) to eliminate a number of worst alternatives, or ii) to choose a number of best alternatives, or ii) to rank the alternatives. In the problem of elimination or choice, the order between the eliminated or chosen alternatives could be also important. In this case we have a mixed problem of iv) choice and ranking. It must be noted that the consideration of MCDM techniques in the definition of dependability benchmarks is not something new [51] [77]. However, existing proposals limit their purpose to the use of MCDM techniques to make explicit the quality model followed to analyse benchmarking measures. This eliminates uncertainties in the process followed to analyse measures, thus improving its repeatability.

This paper makes an step forward in that direction and exploits the differences existing among various MCDM techniques in order to diversify the analysis process and gain confidence in conclusions. It must be underlined that the approach is not useful for checking the correctness of the analysis process itself. The proposal limits its scope to the comparison of the conclusions issued from applying two different MCDM techniques, attending to the same analysis criteria, despite its correctness, to an existing set of benchmarking measures. By checking the existence of discrepancies in the conclusions, one can detect misuses of MCDM techniques, thus being able to fix existing interpretation errors. Once conclusions issued from the application of MCDM techniques are coherent, one can gain confidence on the consistency of reported conclusions, even if such conclusions are not correct because of a problem in the interpretation of input requirements.

This paper is structured as follows. First, section 5.2 introduces the case study that will illustrate the proposal all through the rest of the paper. It will also exemplify the application of an MCDM technique named Logic Score of Preferences (LSP) to the considered

case study. Then, section 5.3 provides a high level view of the approach, details how to apply an alternative MCDM technique, called Analytic Hierarchy Process (AHP) to the same case study, and describes the process followed to detect inconsistencies between rankings promoted by LSP and AHP techniques. Finally, section 5.4 shows the usefulness of the approach, section 5.5 discusses benefits and drawbacks of the proposal, and section 5.6 closes the paper.

5.2 Case study

Wireless Mesh Networks (WMNs) are a particular type of ad hoc networks which is currently being used, among other things, to provide cheaper and more flexible access to Internet than their wired counterparts to isolated or remote areas. As these networks may be deployed in very different scenarios, they may be subjected to a wide range of perturbations (both accidental faults and malicious attacks). Accordingly, and taking into account that a single perturbation has been considered as the most important for each scenario, the aim of this case study is to determine in which of the five proposed scenarios it could be more interesting to deploy that network. Results will be analysed by means of a multiple-criteria decision-making (MCDM) method, the Logic Score of Preferences (LSP), to score and rank the different considered scenarios.

5.2.1 Experimental set up and results

The considered WMN consists of 16 static nodes deployed as shown in Fig. 5.1. REFRAHN, the *Resilience Evaluation FRamework for Ad Hoc Networks* supporting this experimentation, makes use of real devices as network nodes, but emulates their visibility by packets filtering. So, the experimental platform for this case study comprises 10 Linksys WRT54GL routers (200 MHz MIPS processor, 16 MB of RAM, IEEE 802.11b/g Broadcom BCM5352 antenna) running a WRT distribution (White Russian), and 6 HP 530 laptops (1.46 GHz Intel Celeron M410 processor, 512 MB of RAM, internal IEEE 802.11b/g Broadcom WMIB184G wireless card, 4 Li-Ion cells battery (2000 mAh)) running an Ubuntu 7.10 distribution.

Communications are managed by *olsrd* (www.olsr.org), the most extended implementation of the popular Optimized Link State Routing (OLSR) protocol, in its version 0.5.6. The applicative traffic addressed to exercise the network is defined in terms of synthetic UDP Constant Bit Rate (CBR) data flows of 200 Kbps, similar to those observed in daily scenarios [60]. The workload then consists in three of these data flows being exchanged among network nodes.

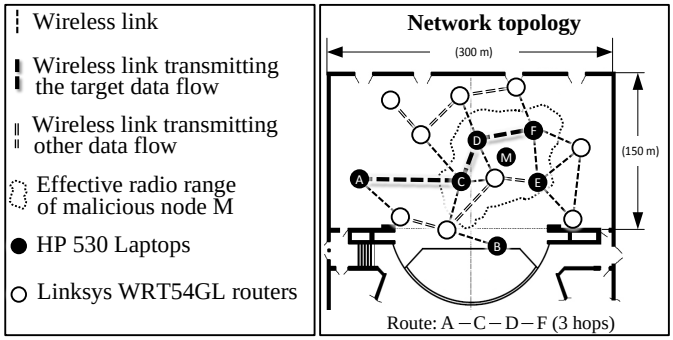

Figure 5.1: Wireless mesh network topology

The perturbations considered in this study (fault- and attack-load) is a subset of the most harmful faults in the domain of WMNs [61]. These perturbations define the five different scenarios considered for this case study: one in which accidental faults, like ambient noise (A), are the most predominant, and the rest where the routing protocol faces various malicious attacks, such as selective forward (S), jellyfish (J), tampering (T), and flooding (F) attacks. Only one of this perturbations will be injected in each of the five considered scenarios, so it could be possible to determine the impact of such particular perturbation on the network. The target data flow for all the considered perturbation is the 3-hop communication between nodes A and F in Fig. 5.1. Whenever a perturbation requires the participation of a malicious node to perpetrate the attack, node M in Fig. 5.1 will play that role.

The set of measures that will be used to characterise the behaviour of the network in the presence of perturbations consists of 5 different measures: i) the average amount of traffic effectively received during experimentation (*throughput*), ii) the average packets delay in milliseconds (*delay*), iii) the percentage of the time the routes are available for inter nodes communication (*availability*), iv) the percentage of packets whose data remain unaltered (*integrity*), and v) the average energy consumed by nodes (*energy*).

For each of the considered scenarios, including a perturbation free one, a total of 15 experiments were executed with a duration of 9 minutes each. The average results obtained from all the experiments performed in each scenario are presented in Table 5.1.

As can be seen, the interpretation of the whole set of results listed in Table5.1 is not straightforward, and multiple-criteria decision-making (MCDM) methods are really helpful to guide the comparison among different scenarios [110]. Our prior research focused on integrating one of these methods, the Logic Score of Preferences (LSP) in particular, into the common dependability benchmarking flow [49] [51].

Table 5.1: Experimental results for each scenario

Scenario	Throughput (Kbps)	Delay (ms)	Availability) (%)	Integrity (%)	Energy (J)
(A)mbient noise	145.2	48.2	73.6	92.12	8.2
(S)elective forwarding	121	42	91.2	97.53	8
(J)ellyfish	184.8	1086.5	88.7	98.54	10.3
(T)ampering	183.6	39.7	93.1	5.2	10.6
(F)looding	149	62.9	72.1	97.56	15.4

5.2.2 Multiple-criteria decision-making: LSP as an example

LSP aims at characterising each system through a single 0-to-100 score which could be used to easily compare and rank eligible alternatives. The final score of the system is obtained by the successive aggregation of intermediate scores according to a defined *criteria tree hierarchy*. Each aggregation takes into account the particular contribution (*weight*) of each subcriterion to the upper level criterion and the intensity of their relation (*operator*). The scores for the base level criteria are obtained by normalising the obtained results according to a given minimum and maximum values (*thresholds*). All these elements, hierarchy tree, weights, operators, and thresholds, constitute the so called *quality model*. The scores for the rest of upper level criteria are computed by using the generalised power mean (see Equation 5.1).

$$score = \sum_{i=1}^{\text{number of subcriteria}} (\text{weight}_i \text{score}_i^{\text{operator}})^{\frac{1}{\text{operator}}} / \sum_{i=1}^{\text{number of subcriteria}} \text{weight}_i = 1 \quad (5.1)$$

The quality model for a given system should be specified prior to experimentation, so its definition is not influenced by experimental results. The constituent elements of the quality model should faithfully reflect the requirements the system must meet. This is the available information for the considered case study: *"The main concern of the deployed WMN focuses on the dependability of supported communications, as sensitive information that should not be altered will be exchanged among network nodes. Thus, preserving the integrity of exchanged packets is of primary importance, whereas the availability of the routes although still of interest is a secondary matter. The network performance also contributes to provide a good quality service, but is not as much important as its dependability. Increasing the network throughput is the main priority to increase the network performance, whereas the delay of the packets is not so important as long as they finally reach their destination. As the nodes of the network will be continuously powered, reducing their energy consumption can be considered as a nice bonus, but not a strong requirement."* Taking all this into account, the quality model reflecting our criteria and optimisation goals for the considered WMN is depicted in Fig. 5.2. It must be noted

that, although not included for space constraints, the thresholds used to normalise the measures, and the required normalisation functions, should also be extracted from the requirements, existing literature, or practical experience.

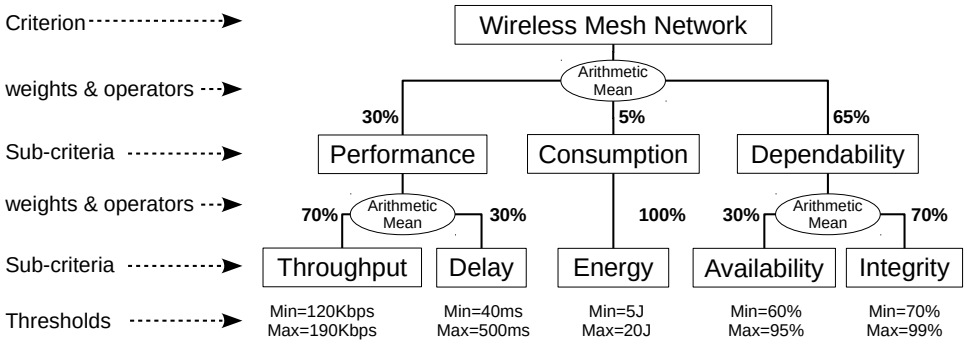

Figure 5.2: LSP quality model for the considered case study

After applying the proposed quality model to the results listed in Table 5.1, the scores obtained for each of these scenarios lead to the following ranking (from best to worst): J (83.44), S (73.83), F (68.76), A (62.61), T (49.65). Accordingly, the considered WMN is best suited to be deployed in scenarios facing jellyfish and selective forwarding attacks, whereas it is not really usable in scenarios facing flooding attacks.

5.2.3 Limitations of the approach

As shown, MCDM methods, like LSP used in this case study, are powerful tools to ease the comparison among different alternatives to select that optimising the defined criteria. However, some values of the multiattribute decision models, like the weights, operators, and thresholds in the quality model presented in Fig. 5.2, are often subjective (judgemental). Lack of precision and accuracy when specifying the requirements of the system (criteria and goals), like the vague natural language description presented in this case study, the misinterpretation of these requirements, or their mapping into quality model attributes, are very important sources of uncertainty. As final rankings provided by MCDM methods are sensitive to changes in their input parameters [39], those uncertainties may lead to very different decisions.

Accordingly, the question of which is the level of confidence that can be placed on the ranking provided by MCDM methods when applied to dependability benchmarks arises. Any variation in the quality model attributes, either due to misinterpretations or vague specifications, may result in wrong decisions which may greatly comprise the depend-

ability of target systems. Hence, the provision of mechanisms to detect and even diagnose any potential inconsistency in the ranking obtained via MCDM methods is indispensable to increase our confidence on provided conclusions.

5.3 Proposal

The main problem once conclusions are provided by the selected MCDM method is that there is no way of determining whether they are right, or at least they seem coherent, taking into account the existing sources of uncertainty in the definition of the required quality model. However, in this context, techniques like *back-to-back testing* may prove useful to detect and possibly diagnose potential flaws in the conclusions obtained.

Back-to-back testing involves cross-comparison of all responses obtained from functionally equivalent components [114]. If any of the comparisons signals a difference the problem is further investigated and, if necessary, a correction is applied. Translating this approach into the considered dependability benchmarking context involves i) applying different MCDM methods to analyse the results issued from experimentation, and ii) comparing the provided rankings to detect existing inconsistencies. If those rankings are coherent, although their correctness cannot be completely guaranteed, the confidence that can be placed on them highly increases. In concrete, this case study promotes the use of a MCDM method called *Analytic Hierarchy Process* (AHP) [89], in parallel with LSP, to achieve this goal.

Next sections describe in detail the AHP technique and the process defined to find out any meaningful dissimilarities between LSP and AHP conclusions.

5.3.1 AHP as an alternative MCDM

AHP [89] is a MCDM method that, instead of a final score, provides a *priority* for each considered alternative reflecting its contribution to the goals optimisation. It shares procedural similarities with LSP, as it also makes use of a hierarchical quality model to aggregate subcriteria into higher level criteria. Accordingly, the very same criteria tree hierarchy may be used for both LSP (see Fig. 5.2) and AHP (see Fig. 5.3) techniques, although the parameters used to characterise the model are not exactly the same and are determined in a different way. This similarity will later ease the comparison between final rankings.

As Fig. 5.3 depicts, the AHP quality model just considers the contribution of each subcriterion to the upper level criterion through their relative *priorities*. These priorities are obtained by means of the pairwise comparison of all the subcriteria contributing to a

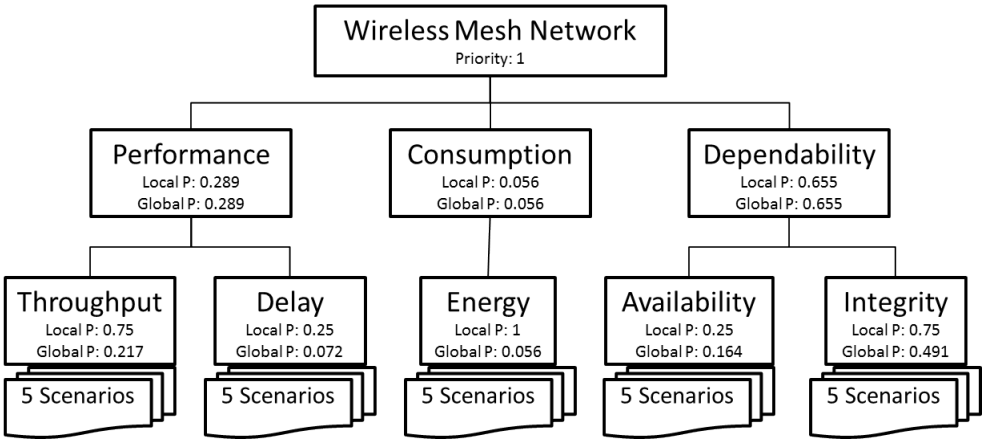

Figure 5.3: AHP quality model for the considered case study

given criterion. Those comparisons are assigned a number (*intensity*) stating how many times more important or dominant one criterion is over another regarding the criterion with respect to which they are compared. Table 5.2 [91] lists the different values (from 1 to 9) denoting the intensity of the importance of criterion A with respect to criterion B.

Table 5.2: The fundamental scale of absolute numbers for pairwise comparison

Definition	**Description**	**Intensity**[a]
Equal	A and B are equally important	1
Moderate	A is somewhat more important than B	3
Strong	A is much more important than B	5
Very strong	A is very much more important than B	7
Extreme	A is absolutely more important than B	9

[a] Intensities of 2, 4, 6 and 8 can be used to express intermediate values. Very close importance values can be represented with 1.1–1.9.

The pairwise comparison of all the criteria contributing to a given criterion is represented in a matrix form, in such a way that if the intensity of criterion A with respect to criterion B is X, then the intensity of criterion B with respect to criterion A is $1/X$. Table 5.3 shows the resultant matrix for the pairwise comparison of *Performance*, *Consumption*, and *Dependability*, with respect to the *Wireless Mesh Network* according to the requirements expressed in Section 5.2.2. In this case, *Dependability* is considered more important than *Performance*, and absolutely more important than *Consumption*, whereas *Performance* is considered just much more important than *Consumption*. Resulting pri-

orities can be derived from the principal right eigenvector of the matrix. However, a fair estimation can be obtained through a more straightforward procedure that will be used in this case study: i) compute the geometric mean for each row of the matrix, ii) sum up the geometric mean obtained for each row, and iii) divide each geometric mean by the total sum. After applying this procedure, shown in Table 5.3, the contribution of each criterion to the final goal is of 0.29 for *Performance*, 0.655 for *Dependability*, and 0.055 for *Consumption*.

Table 5.3: Pairwise comparison matrix of the main criteria with respect to the goal

Wireless Mesh Network

	Performance	*Dependability*	*Consumption*	Row's GeoMean	Priority
Performance	1	1/3	7	1.326	0.29
Dependability	3	1	9	3	0.655
Consumption	1/7	1/9	1	0.251	0.055
			SUM	4.577	

This procedure is recursively applied to compute the priorities for subcriteria with respect to the upper level criterion. Table 5.4 lists the resulting matrices for *Performance* and *Dependability*.

Table 5.4: Pairwise comparison matrices for the subcriteria with respect to Performance and Dependability

Performance

	Throughput	*Delay*
Throughput	1	3
Delay	$^1/_3$	1

Dependability

	Availability	*Integrity*
Availability	1	$^1/_3$
Integrity	3	1

Finally, the pairwise comparison is performed among the different alternatives to determine their contribution to the base level criteria defined in the tree hierarchy. Table 5.5 lists the resulting matrices for *Throughput*, *Delay*, *Energy*, *Availability*, and *Integrity*.

When defining these matrices, it is important keeping the consistency of the pairwise comparisons. For example, if the intensity of criterion A with respect to criterion B is 3, and the intensity of criterion B with respect to criterion C is 3, then to keep the consistency, the intensity of criterion A with respect to criterion C should be more than 3. The consistency of the pairwise comparison matrices is computed by the so called *Consistency Index* (CI) [89] and, although it will not be described here due to space constraints, all the matrices defined in this case study proved to be consistent.

Table 5.5: Pairwise comparison matrices for alternatives with respect to the base level criteria

Throughput

	A	S	J	T	F
A	1	2	1/3	1/3	1
S	1/2	1	1/5	1/5	1/2
J	3	5	1	1	3
T	3	5	1	1	3
F	1	2	1/3	1/3	1

Delay

	A	S	J	T	F
A	1	1	9	1	3/2
S	1	1	9	1	3/2
J	1/9	1/9	1	1/9	1/9
T	1	1	9	1	3/2
F	2/3	2/3	9	2/3	1

Availability

	A	S	J	T	F
A	1	1/5	1/5	1/5	1
S	5	1	1	1	5
J	5	1	1	1	5
T	5	1	1	1	5
F	1	1/5	1/5	1/5	1

Integrity

	A	S	J	T	F
A	1	2/3	2/3	9	2/3
S	3/2	1	1	9	1
J	3/2	1	1	9	1
T	1/9	1/9	1/9	1	1/9
F	3/2	1	1	9	1

Energy

	A	S	J	T	F
A	1	1	2	2	4
S	1	1	2	2	4
J	1/2	1/2	1	1	2
T	1/2	1/2	1	1	2
F	1/4	1/4	1/2	1/2	1

Priorities must be understood at two levels: *local* priorities to their upper criterion, directly obtained from the defined matrices, and *global* priorities with respect to the goal, computed as the local priority multiplied by the global priority of its upper level criterion. For example, *Throughput* and *Delay* have a local priority of 0.75 and 0.25 respectively, computed from the matrix defined in Table 5.4. This is their priority with respect to *Throughput*. However, their global priority with respect to the final goal is $0.75 \times 0.289 = 0.217$ and $0.25 \times 0.289 = 0.072$, respectively. Fig. 5.3 depicts all the local and global priorities for the defined criteria.

This very same procedure is then applied for the priorities obtained for each alternative with respect to the base level criteria. For instance, Scenario A has a local priority of 0.120 with respect to *Throughput* and a global priority of $0.120 \times 0.217 = 0.026$ with respect to the global goal according to its contribution to *Throughput*. Resulting priorities for each alternative are then added up to obtained their final priority. These priorities are then used to rank the alternatives according to their contribution to the optimisation of the goal. In this case study, the final ranking from best to worst is: J (0.2626), S (0.2252), F

(0.1861), A (0.1632), and T (0.1629). So, the target WMN is best suited to be deployed in scenarios facing jellyfish and selective forwarding attacks, whereas it should not be considered for scenarios suffering ambient noise or tampering attacks.

5.3.2 Detecting inconsistencies in provided rankings

The use of two different MCDM methods enables the comparison of the provided rankings to increase de confidence that can be placed on the provided conclusions. Basically, the rankings obtained by applying the LSP and AHP quality models to the results of the dependability benchmark are compared to check whether they are coherent or not. As both techniques follow a different procedure to compute final rankings any misinterpretation of the requirements, procedural errors, or simple transcription mistakes may probably reflect on the provided output (ranking). But, as both techniques are based on the same criteria hierarchy tree, this enables the possibility of tracking inconsistencies down the tree to look for their origin. So not only potential problems may be detected but, in some cases, also diagnosed. The flow diagram representing the procedure to be followed for the back-to-back testing of LSP and AHP rankings is depicted in Fig. 5.4.

The very first step consists in comparing the rankings for the root of the criteria hierarchy tree (goal). In case that no meaningful inconsistencies are found, then the process ends and the rankings are considered coherent. This is what happens in this case study, as alternative scenarios are sorted as J-S-F-A-T from best to worst by both techniques. It must be noted that very small inconsistencies may appear due to the different nature of the considered MCDM methods. For instance, two alternatives may present very close scores/priorities but take reversed positions in both rankings. This probably does not invalidate the provided rankings, but points out that these alternatives are really so close to optimise the goal that they could be considered as interchangeable. In case that more meaningful inconsistencies are found it is necessary to go down the hierarchy tree to look for their origin.

The rankings for the next level subcriteria are also check for inconsistencies. If no meaningful inconsistencies are found this means that, probably, the problem is related to the weights (LSP)/priorities (AHP) computed for the upper level criterion, which should be checked against the requirements. Otherwise, it is necessary to go further down the hierarchy tree in a recursive way.

Finally, in case that the lowest level criteria are reached, and no discrepancies are found, this probably means that thresholds (LSP)/weights (AHP) are not correctly defined at this level, which should be checked against the requirements. If this check is inconclusive, then the problem is likely related to the function used to normalise the measures (LSP).

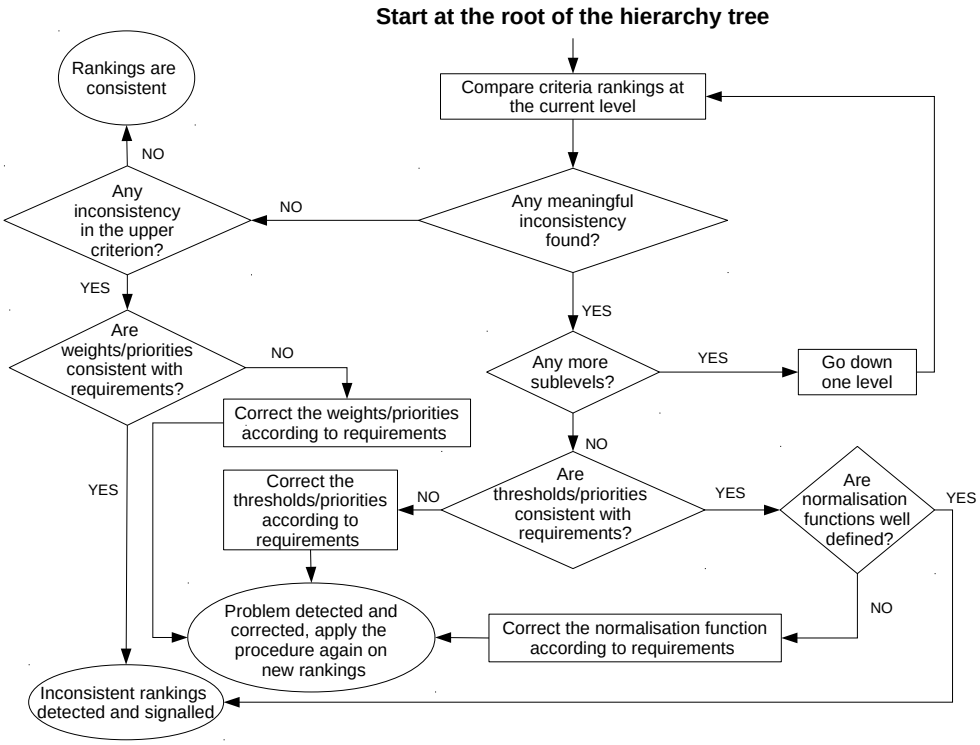

Figure 5.4: Flow diagram for back-to-back testing LSP and AHP rankings

If all these checks are fruitless, then it is not possible to diagnose the origin of the inconsistency to correct it but, al least, a potential problem in the provided conclusions is detected and signalled. Likewise, it is not possible to ensure the correctness of the provided rankings but the confidence that can be placed on its correctness is largely increased.

5.4 Validation of the proposed approach

The proposed back-to-back testing approach to check the consistency of rankings computed from dependability benchmarks results offers a promising procedure to increase the confidence that can be placed on such conclusions. However, it is necessary to determine whether that procedure is robust enough to detect and even diagnose inconsistencies in those rankings derived from different sources of uncertainty when determining the parameters of the defined quality models.

To show the feasibility of this approach, three different LSP quality models have been proposed (M1, M2, M3), in addition to the two original LSP (M0) and AHP models previously defined in this case study. The first new LSP quality model (M1) includes a misinterpretation (or different interpretation) of the requirements specified in Section 5.2.2 in a vague natural language, in such a way that the contribution of *Performance*, *Dependability*, and *Consumption* to the goal is now 0.3, 0.5, and 0.2, respectively. The second quality model (M2) presents a simple transcription error, as the contribution of *Availability* and *Integrity* to *Dependability* has been reversed (0.7 and 0.3, respectively). Finally, another source of uncertainty, related to the definition of the thresholds used to normalise the obtained measures is considered in the third quality model (M3). In this case, the thresholds for the *Delay* have been tighten in excess ([50, 100] instead of [40, 500]). The different scores and priorities obtained by means of all these quality models are listed in Table 5.6. According to these figures, Table 5.7 lists the final ranking provided by these quality models for the different considered criteria.

As the rankings for AHP and M0 have been already proved to be consistent in Section 5.3.2, let us move to comparing rankings for AHP and M1. As Table 5.7 shows Scenarios F and A swap positions in the provided rankings, thus pointing out a potential inconsistency in the defined quality models. Following the proposed diagram flow (see Fig. 5.4), the rankings for the criteria at the next level are also checked. In this case no further discrepancies are found, so the problem should be related to the weights/priorities (pairwise comparison matrices) defined for the highest level of the hierarchy. Whether the parametrisation of one or the other model, or neither of them, faithfully represents the requirements of the system is for the benchmark analyser to decide. Corrective actions at this level are required and new rankings should be compared again.

Great inconsistencies are also found when comparing rankings for AHP and M2, as the worst scenario for AHP is considered the second best for M2. As in the previous example, the rankings for the criteria at the next level are also examined to search for further discrepancies. In this case, the ranking for the *Dependability* criterion also presents inconsistencies. According to the proposed diagram flow, now it is time to check the next (lowest in this case) level of the hierarchy. No discrepancies are found for *Availability* and *Integrity*, so the problem should be related to the weights/priorities assigned at the *Dependability* level. The requirements specified in Section 5.2.2 clearly state that *"preserving the integrity [...] is of primary importance, whereas the availability [...] is a secondary matter,"* so it is easy to determine that the weights for M2 are wrong. After correcting the error, new rankings must also be compared again.

Finally, when comparing rankings for AHP and M3, no inconsistencies can be found according to the proposed diagram flow (see Fig. 5.4). However, the rankings at *Performance* and *Delay* levels present some discrepancies, and the question of whether this approach is really sound arises. It must be noted that, as stated in Section 5.2.3, MCDM

Table 5.6: Scores/Priorities obtained for all criteria after applying the defined LSP/AHP quality models. Scoring differences with respect to the original LSP model (M0) are highlighted in light grey.

Scenario/ Subcriterion		Measure	LSP score			AHP priority	Subcriterion	LSP score			AHP priority	LSP goal score				AHP goal priority
			M0, M1 & M2	M3				M0 & M1	M2	M3		M0	M1	M2	M3	
A	Throughput	145.2	36	36		0.0261	Performance	54.66	54.66	55.2	0.1509	62.61	64.66	52.89	62.77	0.1632
	Delay	48.2	98.22	100		0.0176										
	Availability	73.6	38.857	38.857		0.0096	Dependability	65.05	50.08	65.05	0.1565					
	Integrity	92.12	76.28	76.28		0.0929										
	Energy	8.2	78.67	78.67		0.0169	Consumption	78.67	78.67	78.67	0.0169					
S	Throughput	121	1.428	1.428		0.0140	Performance	30.87	30.87	31	0.1093	73.83	71.86	72.33	73.87	0.2252
	Delay	42	99.56	100		0.0176										
	Availability	91.2	89.143	89.143		0.0482	Dependability	93.194	90.879	93.194	0.2696					
	Integrity	97.53	94.93	94.93		0.1285										
	Energy	8	80	80		0.0169	Consumption	80	80	80	0.0169					
J	Throughput	184.8	92.571	92.571		0.0755	Performance	64.8	64.8	64.8	0.2674	83.44	79.12	79.17	83.44	0.2626
	Delay	1086.5	0	0		0.0020										
	Availability	88.7	82	82		0.0482	Dependability	93.489	86.92	93.489	0.2696					
	Integrity	98.54	98.413	98.413		0.1285										
	Energy	10.3	64.67	64.67		0.0084	Consumption	64.67	64.67	64.67	0.0084					
T	Throughput	183.6	90.571	90.571		0.0755	Performance	93.6	93.6	93.6	0.3215	49.65	54.8	74.24	49.65	0.1629
	Delay	39.7	100	100		0.0176										
	Availability	93.1	94.571	94.571		0.0482	Dependability	28.371	66.2	28.371	0.0936					
	Integrity	5.2	0	0		0.0132										
	Energy	10.6	62.67	62.67		0.0084	Consumption	62.67	62.67	62.67	0.0084					
F	Throughput	149	41.428	41.428		0.0261	Performance	57.51	56.51	51.26	0.1509	68.76	61.83	53.05	66.89	0.1861
	Delay	62.9	95.021	74.2		0.0176										
	Availability	72.1	34.571	34.571		0.0096	Dependability	76.895	52.710	76.895	0.2108					
	Integrity	97.56	95.03	95.03		0.1285										
	Energy	15.4	30.67	30.67		0.0042	Consumption	30.667	30.667	30.667	0.0042					

Table 5.7: Best to worst ranking of considered scenarios. Differences with respect to AHP ranking are in boldface

Quality model		Performance	Dependability	Consumption	WMN
AHP		T-J-F/A-S	J/S-F-A-T	S/A-J/T-F	J-S-F-A-T
LSP	M0	T-J-F-A-S	J-S-F-A-T	S-A-J-T-F	J-S-F-A-T
	M1	T-J-F-A-S	J-S-F-A-T	S-A-J-T-F	J-S-**A**-**F**-T
	M2	T-J-F-A-S	**S**-**J**-**T**-**F**-**A**	S-A-J-T-F	J-**T**-**S**-**F**-**A**
	M3	T-J-**A**-**F**-S	J-S-F-A-T	S-A-J-T-F	J-S-F-A-T

methods are sensitive to input parameters. This means that variations in the input parameters may vary the final ranking. However, this also means that there exist different value ranges for these parameters that do not affect the provided ranking. This is clearly the case of the variation induced in the thresholds for *Delay*. The contribution of the *Delay* to the final goal is not so important, and the dispersion of the measures for each scenario is so small, that the inconsistency is just filtered or masked by the quality model. Obviously, this issue could also be signalled to benchmark analysers, but it seems fairly simpler to make it transparent to them as it really does not affect the final conclusion. The sensitivity of MCDM methods will be further discussed on Section 5.5.

As the considered examples have shown, the proposed approach is able to properly track ranking inconsistencies down the criteria hierarchy tree to find the source of these discrepancies. Hence, back-to-back testing the final rankings provided by MCDM methods prove to be a feasible solution to guide the analysis of dependability benchmarking results and increase the confidence that can be placed on drawn conclusions.

5.5 Discussion

MCDM, as a subdiscipline of operational research, has been supporting decision-making processes for many years in very different application domains. Despite its long tradition, there still exists a recognised fundamental paradox in MCDM. Every single MCDM method claims to offer the best decision but, when different methods are taken into account, not all of them select the same alternative [108]. Accordingly, determining which is the most suitable MCDM method to analyse dependability benchmarking results in a given context could also required the use of another MCDM method, leading to another paradox. One possible option is considering *rank reversals*.

Rank reversals [15] are a particular problem of some MCDM methods which, when subjected to small variations in their inputs or quality model parameters, may produce contradictory rankings. Special tests are usually defined to detect whether this problem

affects the solution provided by a given MCDM method, and alternative methods should be then considered. For example, let us assume that two different routing protocols are being benchmark to select the most suitable to be deployed in a given WMN. Protocol A exhibits more *Throughput* than protocol B, but its overall quality is lower. So, a decision maker could sacrifice the network quality if he considers that is utterly important to obtain the highest possible *Throughput*. However, if a third protocol C is benchmarked, which presents much lower *Throughput* than B but with a very similar overall quality, then the perception of the decision maker may be biased and see protocol B as a more attractive option. As can be seen from the example, rank reversals may also be caused by rational decisions, so they are not always indicative of a faulty decision-making process. Distinguishing whether rank reversals are due to one or the other cause is still a hot topic in the operational research community.

That is why, the back-to-back testing of different MCDM methods to detect any potential inconsistencies in the conclusions provided appears as a sensible option to increase our confidence on final rankings. It is not necessary to determine which is the best MCDM method in absolute terms, but just that provided rankings are consistent with the requirements used to interpret dependability benchmark results. In this proposal, LSP and AHP methods have been selected, as they both can share the same criteria hierarchy tree for their respective quality models. This feature enables the navigation through the different levels of the hierarchy to diagnose the possible source of detected inconsistencies. Obviously, not all MCDM share this feature, but some of them are likely to share other features that could make them compatible to be also used for back-to-back testing. Classifying existing MCDM methods according to shared features, thus enabling the application of different sets of MCDM methods according to, for instance, target application domains, number of considered criteria, or sensitivity to obtained experimental results, is a very interesting topic for further research [121].

As previously mentioned, MCDM methods present different degrees of sensitivity to variations in incoming data or quality model parameters [117]. Those methods with a high sensitivity are more likely to exhibit rank reversal behaviours due to intrinsic sources of uncertainty when defining the quality model and, thus, should not be considered for back-to-back testing when more reliable methods are available. Not so sensitive methods are of great interest, as the uncertainty induced in the quality model (like thresholds, weights, and operators), can be reduced or even masked by the model itself. Accordingly, by estimating the sensitivity of proposed quality models in advance it could be possible to determine and select the least sensitive models, or which parameters should be carefully tuned so as to prevent later inconsistencies. This is also a hot topic requiring further research.

5.6 Conclusions

The work presented here proposes the exploitation of the diversity existing between different multiple-criteria decision making (MCDM) techniques in order to gain confidence on conclusions issued from the analysis of benchmark measures.

The proper comparison of different techniques' results is quite challenging since the diversity existing in the techniques is translated to their related quality models. A concrete approach is proposed to compare the quality model defined by the Logic Score of Preferences (LSP) technique with the one induced, applying the same criteria hierarchy, by the Analytic Hierarchy Process (AHP) method. When rankings issued from both techniques are incoherent, one can detect potential sources of errors in the analysis process and, sometimes, fix them. When they are coherent, one gains confidence on the consistency of drawned conclusions. However, it must be underlined that, despite the coherence of the rankings provided by considered MCDM techniques, conclusions may be incorrect in cases, like where the functional or non-functional requirements of the target systems have been incorrectly captured.

We cannot currently state that this approach applies regardless the couple of MCDM techniques selected for analysis, since their related quality models may exhibit different levels of sensitivity to parameters and input data, which can result in rank reversals. Classifying existing MCDM methods according to their features in order to enable their combined or complementary use attending to aspects relating to the considered application domain, number of criteria, or sensitivity to existing benchmarking measures, remains today an open topic requiring further research. Since the final goal of any benchmark is to drive decisions based on scores and rankings, the final goal of this research is to integrate the use of decision making techniques in the analysis process of dependability benchmarks, something that is today neglected and left in the hands of decision makers acting as benchmark users.

Chapter 6

From Measures to Conclusions using Analytic Hierarchy Process in Dependability Benchmarking

Published at:

- IEEE Transactions on Instrumentation and Measurement

Authors:

- Miquel Martínez[1] - mimarra2@disca.upv.es
- David de Andrés[1] - ddandres@disca.upv.es
- Juan-Carlos Ruiz[1] - jcruizg@disca.upv.es
- Jesús Friginal[2] - jesus.friginal@laas.fr

1. Universitat Politècnica de València, Campus de Vera s/n, 46022, Spain
2. LAAS-CNRS, 7 avenue du Colonel Roche, F-31077 Toulouse, France

Abstract

Dependability benchmarks are aimed at comparing and selecting alternatives in application domains where faulty conditions are present. However, despite its importance and intrinsic complexity, a rigorous decision process has not been defined yet. As a result, benchmark conclusions may vary from one evaluator to another, and often, that process is vague and hard to follow, or even non-existent. This situation affects the repeatability and reproducibility of that analysis process, making difficult the cross-comparison of results between works. To mitigate these problems, this paper proposes the integration of the Analytic Hierarchy Process (AHP), a widely used multi-criteria decision making technique, within dependability benchmarks. In addition, an Assisted Pairwise Comparison Approach (APCA) is proposed to automate those aspects of AHP that rely on judgemental comparisons, thus granting consistent, repeatable and reproducible conclusions. Results from a dependability benchmark for wireless sensor networks are used to illustrate and validate the proposed approach.

6.1 Introduction

Conventional benchmarks characterise computer-based systems attending to different criteria such as performance, power consumption and cost. The aim of any benchmark is enabling the comparison among alternative systems, according to the established criteria, to take a well-based decision. Dependability benchmarks extend this concept to characterise those systems not only in the absence, but also in the presence, of accidental faults and attacks [30]. Accordingly, considered criteria must also encompass dependability and security characteristics [13], like the system robustness against considered perturbations.

In order to be useful, dependability benchmarks must satisfy a number of properties, like scalability, portability, non-intrusiveness, and representativeness. Among them, repeatability and reproducibility are of prime importance. On the one hand, *repeatability*, as defined by the Dependability Benchmarking project [30], guarantees statistically equivalent results when the benchmark is run more than once in the same environment and the same prototype. Without repeatability no one would be able to trust the results obtained from benchmarking experiments. On the other hand, *reproducibility* guarantees that another party obtains statistically equivalent results when the benchmark is implemented from the same specifications and is used to benchmark the same system. Reproducibility is strongly related to the amount of details given in the specifications.

The need to satisfy those properties made that, specially in the early days of dependability benchmarking, lots of works primarily focused on the definition of experimental

procedures to benchmark a wide range of application domains [67], like web servers [37], on-line database transactional systems [112], or automotive systems [87].

All these works place a great emphasis in precisely i) describing the experimental set up, for third parties to be able to reproduce the same experimentation, ii) defining repeatable experimental procedures, including non-intrusive and controllable fault and attack injection techniques, and iii) identifying the set of measures to be considered and how they can be computed from obtained measurements.

It must be noted that little attention is paid to properly characterise measurement systems and express measurement results according to measurement theory [18]. For instance, in the dependability benchmarking domain the terms *measure* and *measurement*, as they will be used throughout this paper, make reference to what is understood as *mesurand* and *measurement result* in the metrology domain [113]. This may negatively affect the repeatability and reproducibility of the experimental procedure due to low quality measurements resulting from incomplete or ambiguous specifications. This problem was addressed in [18] by clearly determining existing sources of uncertainty in dependability measurements for distributed systems, whereas other works, like [8] and [26], focused on improving the quality of dependability measurements.

All these works have greatly contributed to improve dependability benchmarks properties, from specification to experimentation and monitoring. However, the last and also critical stage of the whole process, the comparison of benchmarked alternatives according to obtained measures to make an informed decision is still barely addressed. In most cases, the analysis process is very ambiguous or not documented at all, making the comparison among different results quite difficult. Repeating the same analysis after benchmarking new alternatives, modifying different parameters from the benchmark set up, or just to check the correctness of the previous assessment could lead to very different and even contradictory results due to ill defined analysis processes. In the same way, third parties trying to reproduce the same kind of analysis on their own systems may find it frustrating and meaningless. This lack of explicit criteria to compare alternatives greatly compromises the repeatability and reproducibility of the dependability benchmark process as a whole.

Although the arithmetic and geometric mean are sometimes used to compare alternatives, they are rather simplistic techniques that fail to grasp all the complex relationships existing among selected criteria. For instance, improving one criterion, like *throughput*, may negatively affect another criterion, like *power consumption*. This kind of problem involving conflicting criteria to reach a decision is addressed by *multi-criteria decision making* (MCDM) techniques in the field of operational research [69]. First attempts of using MCDM techniques to make explicit the comparison and selection process in dependability benchmarks were proposed in [77] [76].

This paper takes a step forward in this direction to improve the repeatability and reproducibility of the decision making process of dependability benchmarks by means of MCDM techniques. In concrete, the Analytic Hierarchy Process (AHP) [91], which allows to mathematically express the subjective and personal preferences of an individual or a group when making decisions, has been selected for this study. On the one hand, it has became a widely used technique to solve decision making problems in many areas like business [24], education [101], or engineering [41]. On the other hand, it allows for the hierarchical decomposition of the requirements of the analysis process, which appeals both industry (commonly interested in obtaining the right answer to the problem) and academia (more interested in analysing the problem from different perspectives and levels of detail). Although using a formal method to specify the decision making process, thus making explicit how the comparison and selection process should be performed, AHP requires a number of judgemental decisions relying on the expertise of the evaluator or group of evaluators. Accordingly, the selected alternative may vary depending on several factors that may negatively affect the properties of the dependability benchmark. To prevent this problem, this paper makes a deep analysis of the different elements that may affect the properties of the dependability benchmark and proposes a novel approach, that complements AHP, and that ensures the coherence, consistency, repeatability, and reproducibility of the decision making process.

The rest of the paper is structured as follows. Section 6.2 describes the basis of AHP which are required to understand how they can both benefit and harm the properties of the benchmark. How to integrate AHP into a dependability benchmark is presented in Section 6.3 by means of case study, focusing on wireless mesh networks, which will be used throughout the paper. The different problems deriving from the judgemental decisions taken when applying AHP are identified in Section 6.4. A novel Assisted Pairwise Comparison Approach (APCA) is defined in Section 6.5 to prevent the previously identified problems from affecting the benchmark properties. Finally, Section 6.6 presents conclusions and future work.

6.2 The Analytic Hierarchy Process

The AHP is a technique that enables the decomposition of complex decision-making problems into smaller and easier to solve sub problems, by grouping the different considered criteria into more general criteria. The result is a hierarchical representation of the requirements of the analysis, being the top level criterion (root) the goal of the analysis, and the lowest level criteria (leaves) those defined by the measures to be analysed. Each hierarchy level can be seen as a different level of abstraction of the problem.

This hierarchy will be later used to compute a priority for each considered alternative reflecting its contribution to the goals optimisation. Thus, the contribution of each subcriterion to the upper level criterion is defined through its relative priority. These priorities are obtained by means of the pairwise comparison of all the subcriteria contributing to a given criterion, which eases the task of the evaluator (*law of comparative judgement* [106]). Those comparisons are assigned a number (intensity) stating how many times more important or dominant one criterion is over another regarding the criterion with respect to which they are compared. Table 6.1 lists the different numerical scale (from 1 to 9) [91] denoting the intensity of the importance of criterion A with respect to criterion B. The contribution of each alternative to the lowest level criteria is defined following the very same procedure.

Table 6.1: The fundamental scale of absolute numbers for pairwise comparison

Definition	Description	Intensity[a]
Equal	A and B are equally important	1
Moderate	A is somewhat more important than B	3
Strong	A is much more important than B	5
Very strong	A is very much more important than B	7
Extreme	A is absolutely more important than B	9

[a] Intensities of 2, 4, 6 and 8 can be used to express intermediate values. Very close importance values can be represented with 1.1–1.9.

The pairwise comparison of N elements (criteria or alternatives) contributing to a given criterion is represented in a $N \times N$ matrix known as *pairwise comparison matrix*. As this matrix is reciprocal if the intensity of element A with respect to element B is $I_{AB} = X$, then the intensity of element B with respect to element A is $I_{BA} = 1/X$. Hence, $\forall i, j \in N : I_{ij} \times I_{ji} = 1$.

Those matrices should be consistent, i.e. if element A is more important than element B, and B is more important than element C, then I_{AC} should be greater than I_{BC}. A consistency ratio (CR) can be computed, as detailed in [3], to help evaluators to check that intensities representing the relative importance between elements are consistent. Matrices with $CR < 0.1$ are considered consistent because their small level of inconsistency is due to subjective appreciations [88].

Priorities are derived from the pairwise comparison matrices by computing the principal right eigenvector of the matrix. However, a more straightforward procedure can be used: i) compute the geometric mean (GM) for each row of the matrix, ii) sum up the geometric mean value obtained for each row, and iii) divide each geometric mean by the total sum. Priorities must be understood at two levels. Those directly obtained from the matrix are

known as *local* priorities, and reflect an element's contribution to the immediate upper level criterion. The contribution of an element to the overall goal (*global* priority) is obtained by multiplying the local priority of the element by its upper level criterion's global priority. When a criterion is an immediate descendant of the goal, its local and global priorities are the same.

Finally, the priority of each alternative, i.e. its contribution to the system's goal, is the result of adding all the global priorities obtained for the lowest level criteria. These priorities are the ones defining the final alternatives ranking.

6.3 AHP within dependability benchmarking: wireless mesh networks as a case study

Wireless Mesh Networks (WMNs) are a particular type of ad hoc networks which is currently being used, among other things, to provide cheaper and more flexible access to Internet than their wired counterparts to isolated or remote areas. As these networks rely on a wireless medium with no predefined communication infrastructure to communicate mobile and often performance- and power-constrained nodes, they may be subjected to a wide range of perturbations (both accidental faults and malicious attacks). Hence, in this context, dependability benchmarks aim at assisting network administrators to select the best ad hoc routing protocol for a given deployment, determine the main weaknesses of the selected routing protocol against particular perturbations, and fine tune that protocol accordingly, among other things. What is more, MCDM techniques in general, and AHP in particular, appear as suitable mechanisms to greatly improve the repeatability and reproducibility of the decision making process after including new fault-/attack-tolerance strategies, tuning the selected routing protocol, or injecting a new kind of fault/attack, among other possible uses. Accordingly, the classical stages any dependability benchmark follows have been enriched to integrate the AHP decision making process.

The set of measures defined to characterise the behaviour of the network in the presence of perturbations consists of five different measures: i) the average amount of traffic effectively received during experimentation (*throughput*), ii) the average packets delay in milliseconds (*delay*), iii) the percentage of time routes are available for inter nodes communication (*availability*), iv) the percentage of packets whose data remain unaltered (*integrity*), and v) the average energy consumed by nodes (*energy*). The worst case threshold for each of the considered measures in this case study has been defined as i) 120 Kbps for *throughput*, ii) 300 ms for *delay*, iii) 60 % for *availability*, iv) 70 % for *integrity*, and v) 20 J for *energy*.

At this stage, those measures should be grouped together into higher level criteria to define the required AHP hierarchy. As shown in Figure 6.1, this benchmark considers

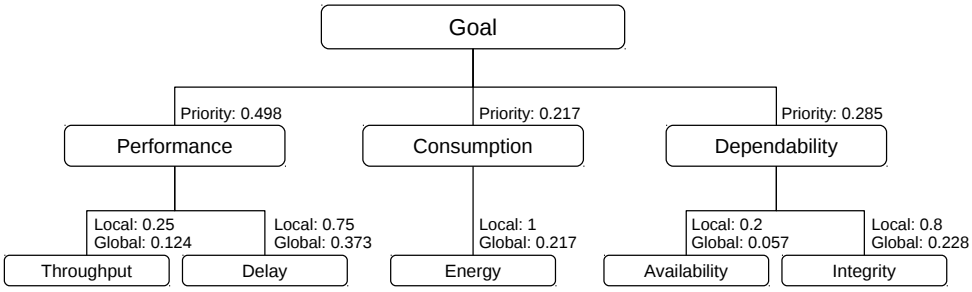

Figure 6.1: AHP hierarchy tree making explicit the analysis criteria

three upper level criteria: *performance*, *dependability*, and *consumption*. The particular contribution of each criterion to the immediate upper level criterion and the goal is computed by means of pairwise comparison matrices. Being the main aim of the network to enable the communication among nodes, the benchmark user has decided that a good performance should be of prime importance. Furthermore, as targeting WMNs, power consumption cannot be considered negligible for mobile devices, and it should be just a little less important than dependability. This is translated into the matrix depicted in Figure 6.2, which also illustrates the process followed to compute the related local priorities. Figure 6.1 shows the resulting local and global priorities for all the considered criteria after the benchmark user has built the required matrices. This makes explicit the relationship among criteria, so any ulterior analysis could be carried out following exactly the very same directives, thus enhancing its repeatability and reproducibility. Furthermore, defining these relationships *before* the experimental procedure takes place ensures that priorities are not biased by prior knowledge of obtained measures.

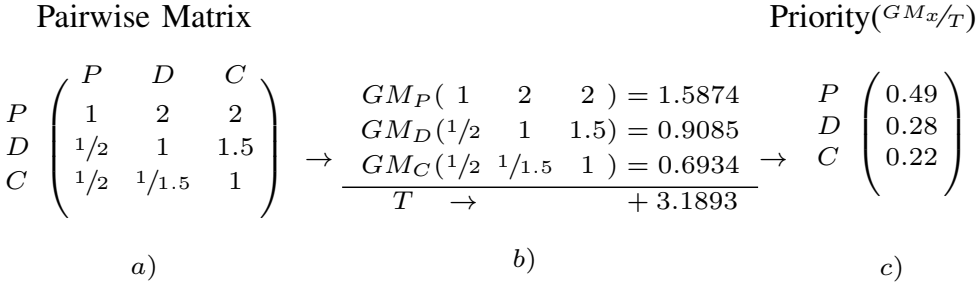

Figure 6.2: Pairwise comparison matrix (a), geometric means (b), and local priorities (c) for performance (P), dependability (D), and consumption (C)

Table 6.2: Experimental set up

Feature	Description
Nodes	10× Linksys WRT54GL routers 6× HP 520 laptops
Routing protocol	Optimized Link State Routing olsrd v 0.5.6
Traffic	UDP Constant Bit Rate of 200 Kbps extracted from daily observations [60]
Perturbations	(A) Ambient noise (S) Selective Forward attack* (J) Jellyfish attack* (T) Tampering attack* (F) Flooding attack*
Target	3-hop communication between nodes A and F in Figure 6.3
Number of experiments	15 per perturbation
Duration	9 minutes per experiment

* Node M in Figure 6.3 plays the role of Malicious node

The particular experimental set up for this case study is listed in Table 6.2. This set up defines five different scenarios in which the target network is subjected to one of the five most harmful perturbations in the WMNs domain [61]. The goal of this experimentation is to compare the behaviour of the network in the presence of these five faults and determine which are the best and worst scenarios for the selected routing protocol. In this way, specific configurations and countermeasures could be deployed to face those weaknesses. The detailed experimental procedure, including how measurements are taken and how they are processed to obtain the required measures can be found in [7].

Table 6.3: Experimental results for each scenario

Scenario	Throughput (Kbps)	Delay (ms)	Availability) (%)	Integrity (%)	Energy (J)
(A)mbient noise	145.2	48.2	73.6	92.12	8.2
(S)elective forwarding	121	42	91.2	97.53	8
(J)ellyfish	184.8	1086.5	88.7	98.54	10.3
(T)ampering	183.6	39.7	93.1	5.2	10.6
(F)looding	149	62.9	72.1	97.56	15.4

The average value of each measure for each scenario is presented in Table 6.3. At this point, obtained measures should be analysed according to the previously defined hierarchy tree and the stated thresholds to rank the considered alternatives. It is now when pairwise comparison matrices for the lowest level criteria (measures) are built. Next section

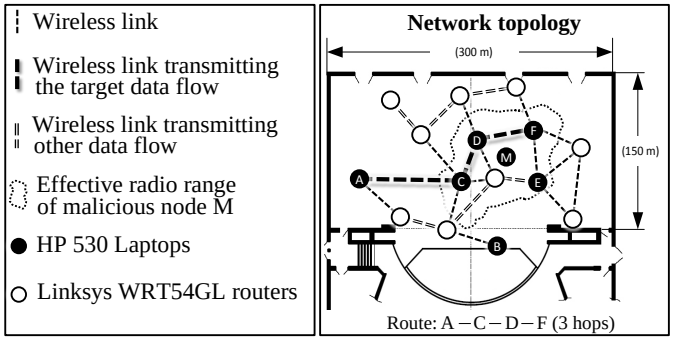

Figure 6.3: Wireless mesh network topology

studies in detail a number of threats to the reliability, reproducibility, and repeatability of the analysis process supported by these matrices.

6.4 Pairwise comparison of alternatives: threats to dependability benchmarking properties

The integration of AHP into dependability benchmarks specification presents clear benefits to the repeatability and reproducibility of the results analysis process. Making explicit the hierarchical aggregation of criteria and their particular contribution to the system's goal ensures that all evaluators will address the decision making problem following the very same guidelines. For instance, i) a given evaluator may benchmark a number of alternatives and apply the same guidelines later when new alternatives are available (new perturbations, new routing protocols, new fault tolerance mechanisms, etc.), ii) another evaluator may later repeat that analysis process on the same data to check the correctness of the procedure and understand the reasoning behind the obtained ranking, and iii) different evaluators may perform new experiments on similar scenarios and they can now follow the same decision making process to compare drawn conclusions.

The application of AHP in the last stage of the dependability benchmarking process (analysis of results), involves the pairwise comparison of alternatives with respect to the lowest level criteria (measures) to compute their local priorities. As this relies on the experience and judgement of evaluators, they are usually selected among experts in the field. Even though alternatives are compared two by two to ease the task of evaluators, and they are experts in their field, this paradoxically poses a number of threats to the reliability, repeatability, and reproducibility of the decision making process. For instance, in case of considering a large number of alternatives and measures, the huge amount of

different comparisons to be made can wear out the evaluator, who can become careless in the following pairwise comparisons. In the same way, if a large number of comparisons is required, they will probably be performed in successive days, which may induced small variations in the evaluator's judgement.

	73.4	88.6	88.7	90.5	71.9
73.4	1	1/3	1/3	1/5	1
88.6	3	1	1	1/2	(3)
88.7	3	1	1	1/2	(3)
90.5	⑤	2	2	1	⑤
71.9	1	1/3	1/3	1/5	1

Figure 6.4: Pairwise comparison matrix for availability as defined by evaluator 1

As an example, Figure 6.4 depicts the pairwise comparison matrix defined by evaluator 1 (Ev_1), in this case study, for the availability of the network. Although the consistency ratio of this matrix indicates that pairwise comparisons are *consistent* ($0.0014 < 0.1$), that does not mean that they are *coherent*. Scenarios S and J with availability 88.6% and 88.7%, respectively, have been considered *somewhat more important* than scenarios A and F, with availability 73.4% and 71.9%, respectively. It may seem coherent to evaluate with the same intensity (3) a difference of 15.2 and 15.3 percentage points (*pp*) with respect to scenario A, and 16.7 and $16.8pp$ with respect to scenario F (differences in a $1.4 - 1.6pp$ range). However, scenario T, with availability 90.5%, have been considered as *much more important* than scenarios A and F. This is clearly not coherent with previous comparisons, as differences of 17.1 and $18.6pp$, in a $0.3 - 1.8pp$ range with respect to previous ones, have been assigned a much higher intensity (5).

A common, although time consuming and costly, approach to minimise all these problems derived from the judgemental nature of pairwise comparisons is inviting a set of experts to take part in the evaluation process. Even though the computed consistency ratio may prove matrices to be consistent, and assuming that they are all coherent, the internal (judgemental) guidelines and comparison scales used by each evaluator renders pairwise matrices very different among evaluators. Figure 6.5, which depicts the matrices built by all five evaluators (Ev_1 to Ev_5) for the energy consumed by network nodes, illustrates this fact.

For instance, the intensity of scenario A with respect to scenario F (I_{AF}) has been defined as 5, 3, 6, 7, and 8 by the evaluators. The same can be said about I_{SF} $(5, 4, 8, 8, 9)$ and I_{JF} $(2, 3, 4, 6, 7)$. The dispersion of these intensities is so large (from equal/moderate importance to very strong importance) that, although the reasoning of a given evaluator can be easily followed, it is very difficult to find a common line of reasoning among all

Ev_1

	A	S	J	T	F
A	1	1	2	2	5
S	1	1	2	2	5
J	1/2	1/2	1	1	2
T	1/2	1/2	1	1	2
F	1/5	1/5	1/2	1/2	1

Ev_2

	A	S	J	T	F
A	1	1/1.5	2	2	3
S	1.5	1	4	4	4
J	1/2	1/4	1	1	3
T	1/2	1/4	1	1	3
F	1/3	1/4	1/3	1/3	1

Ev_3

	A	S	J	T	F
A	1	1/3	2	2	6
S	3	1	4	4	8
J	1/2	1/4	1	1	4
T	1/2	1/4	1	1	4
F	1/6	1/8	1/4	1/4	1

Ev_4

	A	S	J	T	F
A	1	1/4	5	5	7
S	4	1	6	6	8
J	1/5	1/6	1	1	6
T	1/5	1/6	1	1	6
F	1/7	1/8	1/6	1/6	1

Ev_5

	A	S	J	T	F
A	1	1/2	4	4	8
S	2	1	5	5	9
J	1/4	1/5	1	1	7
T	1/4	1/5	1	1	7
F	1/8	1/9	1/7	1/7	1

Figure 6.5: Pairwise comparison matrices for energy as defined by all 5 evaluators

the evaluators for the target system as a whole. In fact, as shown in Figure 6.6 the local priorities obtained for the lowest level criteria are very different among all the evaluators.

Throughput

	$Ev1$	$Ev2$	$Ev3$	$Ev4$	$Ev5$
A	0.096	0.085	0.136	0.130	0.092
S	0.029	0.029	0.041	0.040	0.028
J	0.390	0.383	0.350	0.350	0.400
T	0.390	0.383	0.350	0.350	0.400
F	0.096	0.119	0.123	0.130	0.080

Delay

	$Ev1$	$Ev2$	$Ev3$	$Ev4$	$Ev5$
A	0.201	0.180	0.218	0.194	0.172
S	0.201	0.180	0.218	0.194	0.227
J	0.024	0.025	0.026	0.023	0.024
T	0.447	0.434	0.380	0.493	0.459
F	0.125	0.180	0.158	0.096	0.118

Availability

	$Ev1$	$Ev2$	$Ev3$	$Ev4$	$Ev5$
A	0.076	0.107	0.065	0.094	0.051
S	0.220	0.164	0.269	0.226	0.264
J	0.220	0.202	0.269	0.226	0.230
T	0.409	0.360	0.328	0.384	0.420
F	0.076	0.167	0.068	0.071	0.035

Integrity

	$Ev1$	$Ev2$	$Ev3$	$Ev4$	$Ev5$
A	0.118	0.226	0.110	0.087	0.079
S	0.285	0.248	0.269	0.253	0.285
J	0.285	0.248	0.331	0.445	0.423
T	0.025	0.027	0.025	0.023	0.023
F	0.285	0.252	0.264	0.192	0.190

Energy

	$Ev1$	$Ev2$	$Ev3$	$Ev4$	$Ev5$
A	0.313	0.251	0.223	0.270	0.306
S	0.313	0.412	0.483	0.519	0.452
J	0.150	0.136	0.128	0.092	0.108
T	0.150	0.136	0.128	0.092	0.108
F	0.075	0.065	0.039	0.028	0.026

Figure 6.6: Local priorities for evaluators' decision matrices

It is to note that, for this case study, the particular ranking for each alternative with respect to a given criterion is nearly the same for all the evaluators. They all agree that the best scenarios for throughput, delay, availability, integrity, and energy are J/T, T, T, J, and S, respectively. Likewise, they all agree that the worst scenarios are S, J, A/F, T, and F, respectively. However, the local priorities are so different that, once applied to the hierarchy tree previously defined, the global priority of each alternative, and thus the final ranking is quite different among evaluators (see Table 6.4).

Table 6.4: Resulting ranking after computing the priority for each alternative

Evaluator	**Ranking (from best to worst)**				
	1	**2**	**3**	**4**	**5**
Ev_1	T (0.2628)	J (0.2587)	S (0.1814)	A (0.1601)	F (0.1370)
Ev_2	T (0.2533)	J (0.2436)	S (0.1888)	A (0.1662)	F (0.1481)
Ev_3	J (0.2525)	T (0.2302)	S (0.2241)	A (0.1552)	F (0.1380)
Ev_4	J (0.2678)	T (0.2391)	S (0.2225)	A (0.1563)	F (0.1142)
Ev_5	J (0.2852)	T (0.2592)	S (0.2168)	A (0.1432)	F (0.0956)
AIJ	J (0.2622)	T (0.2500)	S (0.2081)	A (0.1546)	F (0.1252)

This problem is usually addressed by means of group decision techniques, like the aggregation of individual judgements (AIJ) or the aggregation of individual priorities (AIP) [46]. These techniques try to find a consensus among evaluators depending on whether they want to act together as a single unit (AIJ) or as separate individuals (AIP). AIJ builds pairwise comparison matrices by computing the geometric mean of the individual intensities assigned by each evaluator, whereas AIP obtains the global priority of each alternative by computing the geometric mean of the individual priorities computed by each evaluator [97]. These methods lead to more reliable conclusions as they are obtained from a consensus reached among evaluations performed by a group of experts in the field. The ranking obtained by consensus using the AIJ method is listed in Table 6.4.

Nevertheless, involving a set of experts to increase the confidence that can be placed on the results provided by the analysis process also poses some problems. First, the economic impact of hiring a set of experts could be very high, specially when they must be contacted for pairwise comparison again and again after making any change in the system or the experimental set up, like considering different routing protocols, nodes' speed, mobility pattern, traffic, perturbations), or fault tolerance mechanisms. Second, the time required to build the required pairwise matrices can be quite long, as experts will not surely be fully dedicated to just this task. Third, the reproducibility and repeatability of the process may also be affected as judgements may variate along time. Although applying group decision techniques tends to mitigate this effect, when judgements from several experts fluctuate in opposite directions different rankings may be obtained from the same set of data.

Next section presents the proposed approach to increase the reliability, repeatability, and reproducibility of the decision making process in dependability benchmarking without requiring any set of experts, thus also reducing the cost associated to its participation.

6.5 Assisted Pairwise Comparison Approach

Figure 6.7 depicts the relationship between the priority obtained by the most important criterion in a pairwise comparison matrix of just two elements and the intensity defined in that comparison. As priorities resulting from pairwise comparisons matrices of two elements are complementary, if the priority of the most important element is p, the priority of the other element is $1 - p$. It must be noted that small variations for low intensities result in higher priority variations than in the case of considering high intensities. For example, by changing the intensity of the pairwise comparison between criteria A and B from 2 to 3, the priority of A increases (and thus the priority of B decreases) $8.3pp$. However, when increasing the intensity from 8 to 9, the priority of A only increases (and that of B decreases) $1pp$. Fluctuations for very small intensities (from 1.1 to 1.9)

may imply great differences in the resulting priority. That is why, small variations of the judgement made by evaluators when retaking the comparison process may lead to very different results and greatly affect the expected properties of the dependability benchmarking analysis process.

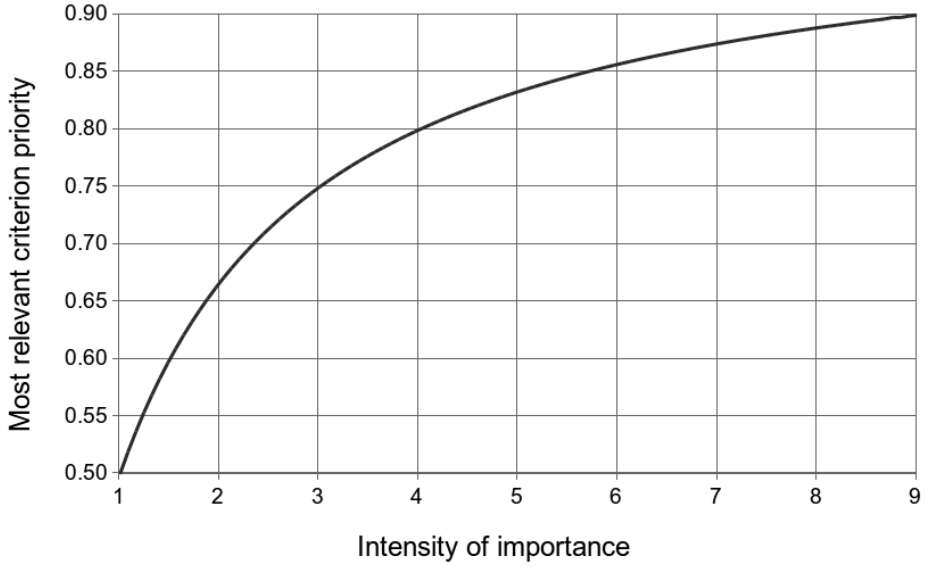

Figure 6.7: Resultant priority from a pairwise comparison depending on the fundamental scale value

The aim of the *Assisted Pairwise Comparison Approach* (APCA) is to automate the pairwise comparison process, thus preventing judgemental decisions from interfering with dependability benchmarking attributes. In such a way, the decision making process becomes completely repeatable and reproducible, as successive applications of this approach always render the very same results. Likewise, it also improves the confidence on the provided rankings, as computed pairwise comparison matrices are always consistent and coherent.

In order to automate the comparison process it is necessary to define a method to unify the interpretation of the relevance of one alternative against another with respect to a given criterion. The first problem is that the values obtained for each measure present very different ranges and hence determining their relative relevance is not so obvious. This issue is usually addressed by normalising those values in a 0 to 100 scale, which states the quality of this alternative with respect to a given measure according to acceptance

values. These acceptance values, the upper and lower thresholds for a given measure, can be obtained by means of experimentation, through literature, or expertise. They must be defined in the dependability benchmark specification, so they could be known beforehand and help any evaluator in understanding and repeating the decision making process. Table 6.5 defines the minimum and maximum thresholds beyond which the quality of any alternative with respect to the selected measure is either maximized (100) or minimized (0).

Table 6.5: Acceptance values determining the required boundaries of the considered measures

Acceptance value	Throughput	Delay	Availability	Integrity	Energy
T_{min}	120Kbps	40ms	60%	70%	5J
T_{max}	190Kbps	300ms	95%	99%	20J

To ensure that the normalisation process is known and applied in the same way, normalisation functions should be also defined in the dependability benchmark specification. This case study makes use of Eq. 6.1 for the linear normalisation of *the higher the better* measures (benefit normalisation function), whereas Eq. 6.2 is the linear normalisation function for *the lower the better* measures (cost normalisation function). These functions compute the quality (q_i) of the value obtained by an alternative (m_i) for a given measure (i) according to its acceptance values (T_{max_i} and T_{min_i}). Table 6.6 lists the quality of the different alternatives for all the considered measures after the normalisation process. Obviously, different normalisation functions (exponential, logarithmic, discrete, etc.) could be defined according to the requirements of the system.

$$q_i = \begin{cases} 0, & m_i \leq T_{min_i} \\ \frac{m_i - T_{min_i}}{T_{max_i} - T_{min_i}}, & T_{min_i} < m_i < T_{max_i} \\ 1, & m_i \geq T_{max_i} \end{cases} \tag{6.1}$$

$$q_i = \begin{cases} 1, & m_i \leq T_{min_i} \\ \frac{T_{max_i} - m_i}{T_{max_i} - T_{min_i}}, & T_{min_i} < m_i < T_{max_i} \\ 0, & m_i \geq T_{max_i} \end{cases} \tag{6.2}$$

After normalisation, the pairwise comparison process can be easily automated, as the difference between qualities is always expressed in pp. The minimum difference that can be found when comparing two qualities is $0pp$, which means that they are exactly equal. Accordingly, it should be associated with an intensity of 1. Likewise, the maximum difference between qualities ($100pp$) can be obtained when one alternative completely satisfies the requirements and the other alternative completely fails to do so. This case should

Table 6.6: Quality of the different alternatives for all the considered measures after normalisation

Scenario	Throughput	Delay	Availability	Integrity	Energy
A	36	96.85	38.86	76.27	78.67
S	1.43	99.23	89.14	94.93	80
J	92.57	0	82	98.41	64.67
T	90.86	100	94.57	0	62.67
F	41.43	91.19	34.57	95.03	30.67

be assigned the highest possible intensity (9). From this analysis is easy to determine that automatically computing the intensities of pairwise comparisons is just a matter of mapping the difference in quality between alternatives to the fundamental scale of comparison. For this case study, APCA considers a uniform distribution of quality difference (0 to 100) along the intensity range (1 to 9). Hence, if Q is the quality difference between two alternatives, being A more important than B, $I_{AB} = (Q \times (9-1) \div (100-0)) + 1$. This means that to increase the intensity in one unit, the difference in quality between alternatives must be of $12.5pp$. Obviously, other distributions can be used according to particular characteristics of the defined dependability benchmark. Algorithm 1 shows the algorithm used to implement APCA and build pairwise decision matrices.

Algorithm 1 APCA

```
1:  n normalized elements to compare
2:  for i from 1 to n do
3:     for j from i to n do
4:        difference = n_i − n_j
5:        if difference < 0 then {n_j is greater than n_i}
6:           a_ji = 1 + (n_j − n_i) × 0.08
7:           a_ij = 1/a_ji {reciprocity}
8:        else if difference > 0 then {n_i is greater than n_j}
9:           a_ij = 1 + difference × 0.08
10:          a_ji = 1/a_ij {reciprocity}
11:       else {n_i is equal to n_j}
12:          a_ij = 1
13:          a_ji = 1
14:       end if
15:    end for
16: end for
```

The consistency of the pairwise comparison matrices generated by APCA has been experimentally verified for scenarios with an increasing number of alternatives (from 3 to

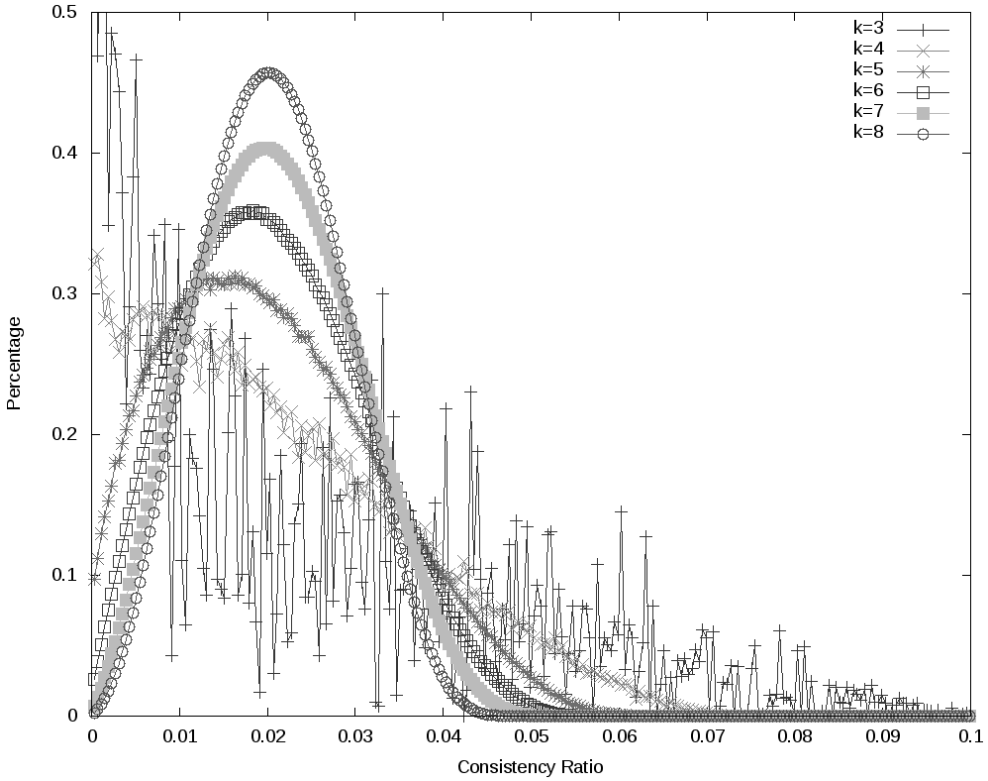

Figure 6.8: Consistency ratio for all possible pairwise comparison matrices with a number of alternatives between 3 and 8

8). For each scenario, the consistency ratio (CR) of all the possible matrices that could be generated with normalised values between 0 and 100 was computed. Figure 6.8 depicts the distribution of the obtained consistency ratio with increasing number of alternatives (k). Matrices were consistent in all cases ($CR < 0.1$).

The application of Algorithm 1 to the normalised values listed in Table 6.6 sets the intensities for the pairwise decision matrices comparing all the alternatives for each of the five alternatives. The local and global priorities for each alternative and the finally obtained ranking are shown in Table 6.7.

It must be noted that this ranking is exactly the same obtained by means of a group of experts using AIJ, which validates the proposed approach. APCA not only provides a deterministic, and thus reproducible and repeatible, analysis process, but also eliminates

Table 6.7: Local/global priorities and ranking obtained by means of APCA

Scenario	Global priorities					Goal	Ranking
	Throughput	Delay	Availability	Integrity	Energy		
A	0.036	0.025	0.004	0.027	0.068	0.160	4
S	0.011	0.025	0.013	0.065	0.068	0.181	3
J	0.146	0.003	0.013	0.065	0.033	0.259	2
T	0.146	0.056	0.023	0.006	0.033	0.263	1
F	0.036	0.016	0.004	0.065	0.016	0.137	5

the costs associated to group decision making techniques as no experts are required for its application.

6.6 Discussion and Conclusions

The same results have been analysed following the requirements defined in Section 6.2. The same hierarchy and priorities for the criteria of the hierarchy have been used by all five evaluators, by the consensus analysis, and by the APCA. The difference between them resides in the process of establishing the intensities in the comparison of alternatives against criteria, where the subjectivity in the analysis is introduced. As illustrated in Section 6.4, the rankings obtained individually by the five evaluators vary on determining which scenario was the best one. However, if the results obtained from the use of APCA are compared with those obtained from a consensus analysis using the GMM, it can be seen that both rankings are equal. As consensus matrices are used in the literature to provide more reliable conclusions, and the conclusions derived with APCA are the same, it can be stated that the conclusions obtained with APCA can be considered correct.

Performing the AHP together with the APCA, automates the process of the analysis of results. Then, evaluators interaction with the results remains limited to defining the hierarchy and the normalization functions, and determining the thresholds for the criteria that are going to be evaluated. Limiting the interaction of the evaluator to the definition of these features, prevent the process from being affected by the subjectivity introduced by the evaluator in the comparison of alternatives versus criteria. At the same time, it guarantees the repeatability of the analysis, as measures will be compared using the same comparison scale. Thus, by simply reproducing the features of the APCA defined in an analysis, external evaluators can repeat the exact same analysis process and compare their conclusions with those extracted from other works. Hence, if the same results are analysed, obtaining the same conclusions is granted, as APCA preserves the repeatability in the analysis.

In addition to that, the use of APCA assures that the decision matrices will be consistent in all situations, and thus do not need to be corrected. Currently there exist different methods that can be applied to improve the consistency of already performed comparison matrices [72][19][88], but even though, evaluators still need to perform them. In this example, five evaluators have performed individual analysis of the results, but relying on multiple evaluators to make the conclusions more reliable, is a hard (sometimes impossible) task. Nevertheless, with APCA one evaluator can perform a more reliable analysis of the results by just defining the required features for APCA, and no time needs to be spent on performing comparison matrices. So, the application of APCA is more straightforward, and less time and effort is required.

Providing the analysis of results in dependability benchmarking with some guidelines is becoming necessary to make it possible to understand the process that lead to a conclusions. In this work we have presented an approach to assist the integration of a well known MCDM technique to perform the analysis process in dependability benchmarking. Its application in this domain can ease the context-aware comparison of systems, thus providing more context oriented conclusions. Nevertheless, this work is a first step to integrate MCDM techniques into dependability benchmarking, and there are still certain aspects, like those regarding the definition of requirements to derive the hierarchical decomposition of the analysis, that need to be part of further research.

Chapter 7

Assessment of Ad Hoc Routing Protocols for Network Deployments in Disaster Scenarios

Published at:

- Workshop on Innovation on Information and Communication Technologies, ITACA-WIICT

Authors:

- Miquel Martínez[1] - mimarra2@disca.upv.es
- Yusheng Ji[2] - kei@nii.ac.jp
- David de Andrés[1] - ddandres@disca.upv.es
- Juan-Carlos Ruiz[1] - jcruizg@disca.upv.es

1. Universitat Politècnica de València, Campus de Vera s/n, 46022, Spain
2. National Institute of Informatics, Tokyo, Japan

Abstract

The use of ad hoc networks to recover the communications infrastructure after a natural disaster situation has been deeply studied in the literature. In these kind of situations, the current infrastructure that provides access to the Internet can be damaged, which turns out in a complete loss of service in some areas. This problem has been approached by creating an ad hoc network conformed by the remains of the damaged infrastructure and additional relays. Thus, providing service to isolated areas. However, the harsh conditions of these deployments and the unstable conditions of the environment can affect its performance. In this work, the deployments done in previous works using real data over the city of Tokyo, are evaluated in the presence of different ad hoc routing protocols to determine the performance of the network. These results are analysed using *multi-criteria decision-making* methods in order to determine under which protocol the network performs the best.

7.1 Introduction

The frequency of earthquakes in Japan is extremely high compared to other countries of the world. Around 30% of the world's earthquake every year take place in Japan. When such a disaster occurs, keeping the population informed is of primary importance, and for that to happen communication infrastructures must keep working.

Nowadays people have quick access to the information through their smartphones or laptops via the Internet. Online news and social networks let the users be aware of the current situation of the disaster, the status of evacuation areas and shelters, or stay in touch with their loved ones.

Communication infrastructures, however, can be damaged during a disaster leaving people in some affected areas isolated from the rest. This lack of communication in some areas is mainly caused by the malfunctioning of the base stations that operate in such areas providing Internet access for the people. But after a disaster, one of the main reasons for these base stations to fail is power outage. Although base stations can be powered with additional sources to keep them running, like power generators or batteries, these are temporary solutions and have a short lifetime.

The use of wireless and infrastructureless technologies, like ad hoc networks, is a common approach in these works. Some of them are focused on determining the kind of ad hoc network to deploy (MANET, VANET, MESH, etc), and the wireless technology to use for the nodes. Others, like [71], approach the connectivity problem by studying how to deploy static and mobile relays to create an ad hoc network to provide Internet access to isolated areas in a disaster scenario in the city of Tokyo (Japan). To do so, this work

applies an algorithm developed in [20] and adapt it to study to what extent the number of nodes can be reduced while providing network service to all nodes.

There are various factors that will impact the performance of this network. As stated in works like [50] [52], an important factor that should be considered when deploying an ad hoc network is the routing protocol used. Thus, being able to determine which routing protocol improves the performance of the network in such harsh conditions can make a difference.

Given that, the work done in this paper focuses on benchmarking the performance of the network deployments done in [71] for different ad hoc routing protocols. These deployments are simulated for a post-disaster scenario in the city of Tokyo. Thus, the benchmarking process will determine which routing protocol improves the performance of the ad hoc network deployed.

Assuming the context for these deployments, a post-disaster scenario, the main purpose of the network is to keep people informed, so having a low ratio of *packet loss* will be more important than having a high *throughput*. This requirements must be reflected in the analysis of results in order to provide meaningful and context-aware conclusions. To cope with this kind of analysis, previous works have shown the feasibility of *multi-criteria decision-making* (MCDM) [63] methods to consider evaluator's requirements in the analysis. So in this work, MCDM methods are used to assist the analysis of the results obtained from the experiments, and provide the reader with our conclusions.

The rest of the paper is structured as follows. Section 7.2 presents a brief description of the ad hoc routing protocols that are evaluated in this work. The experiments performed in this work are described in Section 7.3, while the MCDM method used to analyze the experiments' results is presented in Section 7.4 together with the analysis done. Finally, Section 7.5 discuss the impact of the work done and concludes the paper.

7.2 Routing Protocols

Ad hoc network deployments in disaster scenarios are expected to have the best possible performance, as people lives may rely on this network to work. Thus, this work is a first approach to study the improvement in the network's performance that one routing protocol has over another. However, the number of existing routing protocols is quite large, so as a first approach, among all of them three well known routing protocols have been considered for this work.

OLSR The *Optimized Link State Routing* protocol ([102]) is a proactive protocol that makes use of link-state information to determine the best route from source to

destination among those available. Its constant transmission of *hello* and *topology control* packets let the nodes discover and propagate the information about the status of the network topology in real time. Thus when information has to be forwarded towards a given node, the route is already known. Of course, the constant exchange of information introduces some overhead in the network traffic.

AODV Unlike OLSR, Ad hoc On-Demand Distance Vector Routing ([81]) has a reactive behavior. Route discovery only takes place when information has to be forwarded, not before. When a node has to send information to another node, probe-like packets are sent through the network to find a route for the information to be delivered. This packets are called *Route Request* (RREQ) and *Route Reply* (RREP). This last packet is the response from the destination node, and contains all the necessary information about the route that the information has to follow to reach it. As this packets are only transmitted "on-demand", the overhead introduced in the network traffic is expected to be lower than in OLSR.

DSDV Destination-Sequenced Distance-Vector Routing ([82]) follows a table driven routing scheme based on the Bellman-Ford algorithm. Thus it has a proactive behavior. Each node maintains routing information for all known destinations which is updated periodically. Nodes use a sequence number pattern to announce its routing information to its neighbors. When links are present, even numbers are used, odd numbers otherwise. This way, nodes use this sequence numbers to keep their own routing tables updated and avoid the appearance of routing loops.

In order to evaluate and compare these three protocols it is necessary to define a set of repeatable and reproducible experiments. The conditions of each experiment should be reproduced so the experiment can be repeated using every one of the routing protocols. Next section describes the different aspects that were taken into consideration when defining the experiments.

7.3 Experimental Set Up

The experiments done in this work are based on the deployments studied in [71]. Here, authors study the feasibility of their approach to create an ad hoc network by studying multiple base scenarios. Each scenario differs from the rest in their initial assumption of which areas are disconnected and which ones remain connected. Thus, different initial states led to different deployments of nodes to provide isolated areas with access to the Internet. Given the amount of nodes, all these scenarios were simulated using a widely used tool, the *Network Simulator 3* (NS3) [103].

The upcoming sections provide a description of the process followed to design the experiments and determine the measures that would be used for its later analysis.

7.3.1 Target of evaluation

During a disaster people is encouraged to go to their nearest evacuation site or shelter. The information of their location is available online through governmental sites [43]. These locations are interpreted as areas in Tokyo that may have or not access to the Internet. Thus, they are considered as nodes of the network, meaning that additional nodes must be deployed between them so that every area has access to the Internet, hence creating the ad hoc network. For the experiments, an area of 35 km^2 in the center of Tokyo that encompass a total of 67 sites - nodes, from now on - is considered. The locations of the initial nodes can be seen in Figure 7.1.

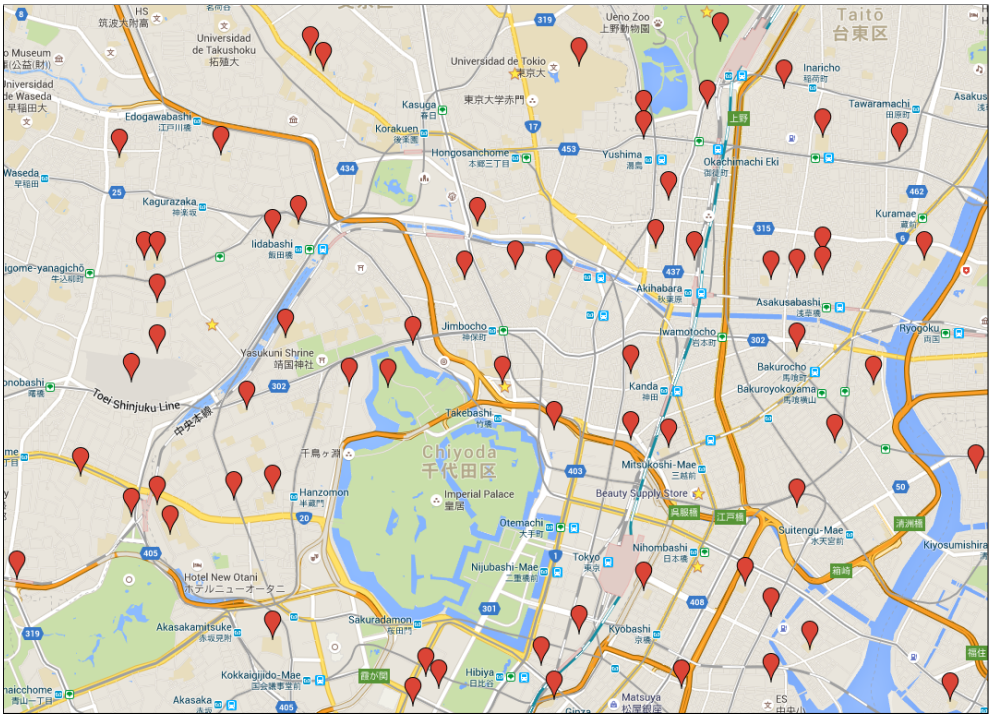

Figure 7.1: Evacuation areas and shelters in the city center of Tokyo.

7.3.2 Definition of the experiments

Determining which of these nodes are initially out of service is done randomly based on the probability that each area has to have access to the Internet after a disaster occurs. This information, as it is described in [71], it is extracted from real data used to create maps of Tokyo that indicate the probability of service in each area in four different time slots (6h, 12h, 18h and 24h) after the disaster. So the four maps must be considered for the experiments, as the probability of having service in an area will drop dramatically between 6 hours and 24 hours after the disaster.

In addition to these four possible initial scenarios, three different workflows are defined to evaluate the performance of the network. Each workflow is defined as a different *Constant Bit Rate* (CBR) that is generated in the disconnected nodes towards the connected ones. The CBRs considered are the following: i) 0.5 Mbps, ii) 1 Mbps and iii) 3 Mbps.

The total number of experiments performed depends also on how many initial scenarios are created. For every map the disconnected nodes are randomly selected, leading to different topologies each time. Then, the more number of initial scenarios are considered, the more number of experiments will be performed. In this work a total of 144 experiments were performed. This number of experiments is the result of considering four initial scenarios for each map, thus leaving 16 topologies to evaluate (4 initial scenarios x 4 maps). Each of this scenarios is evaluated for every CBR defined, which gives a total of 48 different experiments (16 scenarios x 3 CBR). The 48 experiments must be performed considering all three routing protocols defined in previous section, making a total of 144 experiments.

7.3.3 Performance measures

The purpose of this work is to evaluate the performance of the network with different routing protocols. Keeping that in mind, the measures provided as the outcome of an experiment were selected considering the impact they would have on the network performance. The performance of the whole network must be assessed by analysing all the traffic flows generated in the experiment. So, every experiment provides the following information for each data flow in the network: i) The percentage of *Packet Loss*, ii) the *Throughput* perceived by the receiving node, iii) the average *Delay* of the packets, and iv) the *Availability* of a route. This last measure indicates if a route between a disconnected node and an Internet gateway was available when a node needed to send information.

It is obvious that the number of data flows in an experiment can be very high. In order to compare the three routing protocols, the final measures used in the analysis will be the average for all the data flows generated. Table 7.1 shows the final measures obtained for an experiment after its execution with the three routing protocols.

Table 7.1: Average results obtained for an experiment performed using the 12 hours probability map and a CBR of 0.5 Mbps

	Throughput (Kbps)	Delay (ms)	Packet Loss (%)	Unavailable routes (%)
OLSR	418.4	2.3346	8.65	0.3
AODV	418.3	7.661	4.26	0.1
DSDV	404.5	39.686	9.45	0.4

As aforementioned in this paper, when analyzing the results of the experiments it is necessary to do it according to the application context of the network deployment, a disaster scenario. The methodology used to analyze the results from the experiments, as well as the conclusions driven from the analysis are presented in next section.

7.4 Analysis of results

In the benchmarking literature, it is common to find that the analysis of results is done using methods that consider all measures equally relevant, like the *arithmetic mean*, the *geometric mean* or the *kiviat diagrams*. In contrast, some later works in this field ([76][77]) have proved the feasibility of MCDM methods to analyze results considering different levels of importance for each measure, as it happens in real situations. Then, in this work, a widely used MCDM method in different fields of research has been chosen to perform the analysis, the *Analytic Hierarchy Process* (AHP) [91].

The AHP allows the evaluators to model their analysis requirements into a hierarchical structure, where some criteria represent the result of other sub-criteria. The contribution of each criterion to its immediate upper criterion must be quantified, which can be understood as the weight that the value of one criterion has to calculate the value of its upper one. For evaluators to quantify their requirements, pairwise comparisons of the criteria at the same level must be done. Thus, with the use of a numeric scale, evaluators compare the criteria two-by-two to quantify the importance that one criterion has with respect to another to achieve the immediate upper criterion. This process is repeated until every criterion has been compared with all the others. These comparisons are stored in a matrix form, and the final priority of each criterion is calculated by applying the principal right eigenvector of the matrix.

After comparing the four measures considering the application context of the deployment, the priority (or weight) of each criterion (measure) to calculate the global score for each protocol are as follow (in [0,1] scale): 0.09 for *Throughput*, 0.11 for *Delay*, 0.52 for *Packet Loss* and 0.28 for *Unavailable routes*. Figure 7.2 depicts the hierarchical model

of the requirements used in this work to compare the routing protocols and the calculated weights for each one.

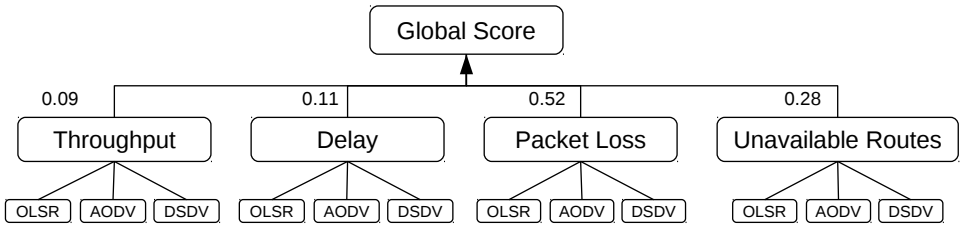

Figure 7.2: Hierarchical model of the requirements applied to analyze the results.

When the priorities for all criteria are calculated, the results obtained from all routing protocol must be pairwise compared too according to each criterion. This will provide a value for each protocol relative to the others in each criterion, so a global score can be calculated for each one of them, thus allowing to rank the protocols from best to worst. However, this process can be very tedious given the fact that 48 different types of experiments were done. To automate this process, the *Assisted Pairwise Comparison Approach* (APCA) (introduced in [78]) was used to compute all the comparisons.

After analysing and comparing the results obtained from performing the 48 types of experiments with the three routing protocols, we found ourselves with 48 rankings. These rankings represent under which protocol the network presented a better performance, so routing protocols are ranked from best to worst.

To determine which of them was the best candidate to be used for a network deployment in this scenario, the accumulated frequency of their ranked positions was studied. These frequencies are depicted in Figure 7.3. It can be seen that even though OLSR and DSDV are ranked in the first position more times than AODV, AODV is ranked around 75% of the time second and the other 25% first. But unlike OLSR and DSDV, AODV is never ranked third. Based on these results, it can be stated that in average, the network will perform better running AODV than with OLSR or DSDV, as these two would present the worst performance in around 40% and 55% of the scenarios, respectively.

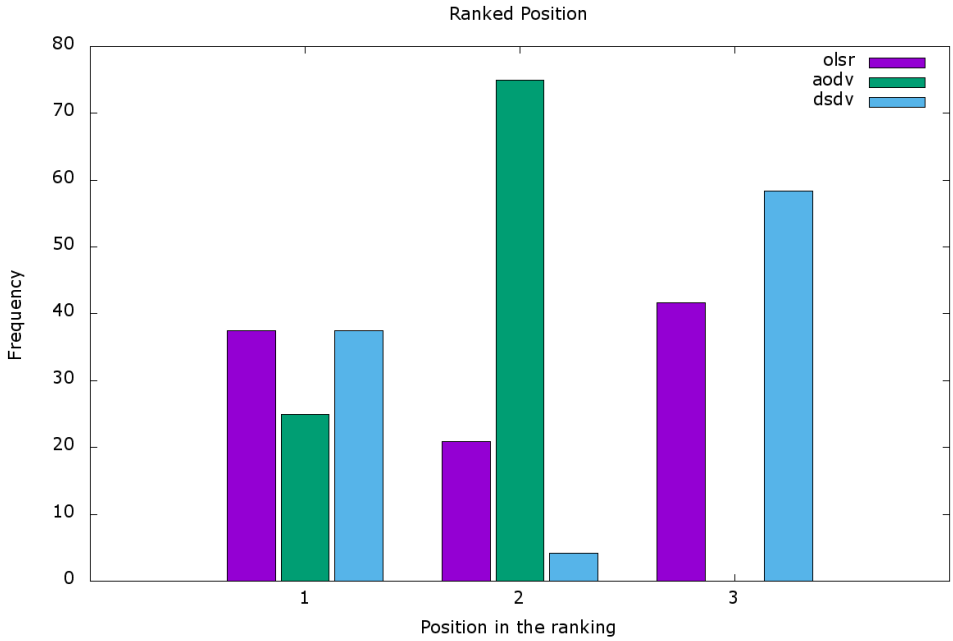

Figure 7.3: Frequency of each routing protocol classified as first, second or third among all experiments

7.5 Conclusions

Natural disasters seem to have become more common during the last decades. In Japan, where earthquakes represent a high threat to population, a lot of measures have been taken and actuation plans have been developed to improve people's safety. However, there are currently no implemented solutions to prevent communication's infrastructure to keep working after a disaster occurs, leaving people without service or Internet access.

Authors in [71] have approached the problem of communication failures through the deployment of an ad hoc network using static and mobile relays. They make use of an algorithm to determine, based on the isolated areas, which is the best way to deploy the network nodes so every isolated area has service again.

Starting from this approach, this work has been focused on evaluating the performance of the ad hoc network deployed with three well known routing protocols, OLSR [102], AODV [81] and DSDV [82], thus find out which one is more suitable for the network. Every protocols was evaluated for a total of 48 different possible deployments generated through the methodology described in [71].

Although traditional approaches like *arithmetic* or *geometric* mean are used in benchmarking studies to analyze the results, the post-disaster scenario of these deployments required a different approach. As people's life may rely on these networks to work properly, the analysis of the experiment's results has been carried out using *multi-criteria decision-methods*, that let evaluators apply specific requirements to the analysis. More concretely, the *Analytic Hierarchy Process*.

This study has shown that the selection of the ad hoc routing protocol can make a significant difference in the performance of the network. From the considered protocols, the network has shown on average a better performance when using AODV than with the other two.

Nevertheless, this study is only a first approach to contribute to the improvement of the performance in this kind of deployments. The next step in this work involves not only to evaluate more routing protocols, but also to compare different deployment techniques. Thus, being able to determine which kind of deployment and ad hoc routing protocol would be better to cope with failures in the communications infrastructures in this type of situations.

Chapter 8

A Multi-criteria Analysis of Benchmark Results With Expert Support for Security Tools

Published at:

- Under preparation for submission to a journal

Authors:

- Miquel Martínez[1] - mimarra2@disca.upv.es
- Nuno Antunes[2] - nmsa@dei.uc.pt
- David de Andrés[1] - ddandres@disca.upv.es
- Juan-Carlos Ruiz[1] - jcruizg@disca.upv.es
- Marco Vieira[2] - mvieira@dei.uc.pt

1. Universitat Politècnica de València, Campus de Vera s/n, 46022, Spain
2. CISUC, Department of Informatics Engineering, University of Coimbra, Portugal

Abstract

The benchmarking of security tools is endeavored to determine which tools are more suitable to detect system vulnerabilities or intrusions. The analysis process is usually oversimplified by employing just a single metric out of the large set of available metrics. Accordingly, the decision may be biased by not considering relevant information provided by neglected metrics. This paper proposes a novel approach to take into account several metrics, different scenarios, and the advice of multiple experts. The proposal relies on experts quantifying the relative importance of each pair of metrics towards the requirements of a given scenario. These judgments are aggregated using group decision making techniques, and pondered according to the familiarity of experts with the metrics and scenario, to compute a set of weights accounting for the relative importance of each metric. Then, weight-based multi-criteria-decision-making techniques can be used to score the benchmarked tools. The usefulness of this approach is showed by analyzing two different sets of vulnerability and intrusion detection tools from the perspective of multiple/single metrics and different scenarios.

8.1 Introduction

Security tools have a growing importance nowadays to help developers protecting their systems against security threats [99]. The usefulness of these tools is many fold, as they may be applied during the development to recommend the best coding practices, during verification and validation phases to disclose vulnerabilities, or after deployment to protect the system against security attacks [99]. The lack of expertise usually leads development teams to trust the outputs of those tools, but research and practice show that their effectiveness is not always satisfactory [33, 115].

Benchmarking is the 'go to' technique when it comes to the assessment and comparison of tools according to some characteristic [56]. Benchmarks were successfully applied in comparing the performance of systems [56], but other benchmarking approaches have been proposed to evaluate other types of properties such as dependability [67]. *The key for the success of the benchmark is the adoption by the community*, and therefore, it is imperative that benchmarking proposals respect a group of properties and provide benchmarking users with useful insight. For this, *one of the most important points is the quality of the metrics used*.

In benchmarks that follow measurement-based approaches, the metrics are computed from the measurements obtained during the benchmark execution [56, 67]. The produced values must be understood in relative terms, and they are mostly useful for comparison or improvement and tuning. Besides, the metrics should respect a set of properties to be

useful for the benchmark users [65, 95]. *A good metric should provide repeatable and reproducible results, be consistent (i.e. should not be open to subjectivity), understandable by the user, and meaningful in the context where it is being applied* [65].

The work on assessment and comparison of security tools resulted in several approaches in recent years (e.g. [9, 33, 40, 115]). These works characterize the effectiveness of the tools in terms of the true positives and false positives, from which general purpose metrics such as precision, recall, and F-Measure [83] are derived. In most of the cases, tools are simply compared using one of these metrics. But even in the cases where different sets of metrics were taken into account, the fact is that, only one of them was finally considered, while the remaining were simply acting as tiebreakers or disregarded [10]. This simplification *bias the conclusions by leaving out the information potentially provided by the ignored metrics*.

Another concern transversal to most benchmarks is that, although they consider multiple metrics, they only rely on a single set of them that should be used in all cases. However, different benchmarking campaigns may have different objectives. So, as the same set of metrics may not be optimal for all the scenarios in a given domain, we argue that, in addition to consider multiple metrics, *benchmarks should also consider the influence that each analysis scenario may have in the relative importance of each metric*.

Finally, *domain experts must be in charge of weighting up the relative importance of each metric within the context of each specific scenario*. This applies even when no experts are explicitly involved in the analysis of benchmarking results. In that case, benchmark users are implicitly assuming the role of experts when ranking and comparing the considered alternatives. If they are not able to demonstrate an acceptable level of expertise, then their analysis can be put in question and it will not be useful in practice.

This paper addresses the aforementioned challenges by proposing **a new analysis approach suitable to weight up the relative importance of benchmark metrics in each application scenario while taking into consideration the opinion of experts**. The approach is called MABRES, that stands for *Multi-criteria Analysis of Benchmark Results using Expert Support*. Compared with existing analysis techniques, the key novelties of MABRES are three-fold. First, it enables various experts to participate in the analysis process, thus providing means to aggregate their individual judgements. Second, it allows the simultaneous consideration of multiple metrics, while enabling traceability from metrics to scores and vice-versa. And third, it considers the influence of scenarios in the interpretation of metrics, thus making the resulting analysis process context-aware.

The outline of this paper is as follows. Section 8.2 details the context of the present research. Section 8.3 presents MABRES and Section 8.4 exemplifies its usefulness

through the *benchmarking of two different types of security tools*. Section 8.5 discusses the pros and cons of the contribution. Finally, Section 8.6 presents conclusions.

8.2 Research Context

Benchmarks are standard tools that allow evaluating and comparing systems or components according to specific characteristics (e.g. performance, dependability, etc.) [56]. It is well known that a benchmark must be defined targeting a particular domain, as this influences the definition of the benchmark components. Although some benchmarks may include other components, the key ones usually are: the **metrics**, a **workload**, a **procedure**, and an **experimental setup**.

Above all, the usefulness of a benchmark is tied up with the metrics used to portray the characteristics of the system and how they provide useful insight according to the objectives of the benchmark user. However, research and practice show that the *currently used approaches to analyze metrics for computer benchmarks are not adequate* [9]. Most benchmarking approaches use a **single metric**, which provides a limited view of the results, or a **small set of metrics**, which does not solve the problem as metrics are normally analyzed in a disjoint manner. This raises the need for ways to combine metrics in order to reflect an aggregate view of system characteristics.

Which metrics should be considered, which scenarios are more interesting for the analysis of such measures, and more importantly, is it really necessary a new method to score evaluated tools attending to such metrics, are some of the relevant questions that thus need to be addressed.

8.2.1 Multiple Metrics for Benchmarking Security Tools

Vulnerability and intrusion detection tools can be seen as binary classifiers, as they classify parts of the target application in one of two classes: vulnerable or non-vulnerable. Several metrics are available to portray the effectiveness of binary classification algorithms or tools, including information retrieval systems and machine learning algorithms [44]. Most of those metrics are computed from raw measures reporting a confusion matrix, which represents the possible outcomes for each classified instance [98]. Such outcomes are basically specified in terms of the amount of *true positive*, *true negative*, *false positive* and *false negative* detections carried out by each evaluated tool.

The number and variety of metrics that can be derived from the aforementioned outcomes is quite important and, despite their distinct denominations in different areas, many of them are in practice synonyms. According to the precise meaning of available metrics

and the feedback provided by 21 experts, the research carried out in [10] proposed a list of 5 representative metrics for characterizing the various attributes of interest when benchmarking security (vulnerability detection) tools. The list can be found in Table 8.1. **Recall** determines the ratio of reported positives that are correct among all the existing vulnerabilities, while **precision** indicates from all the reported positives, which proportion of them are correctly classified. **F-measure** is the harmonic mean between recall and precision. **Informedness** and **markedness** were defined in [83] as a way to measure the accuracy of a predictor (a tool in this context) considering chances of doing right predictions based on the number of vulnerabilities. Informedness combines recall and its inverse metric to express in a single measure how informed are the classifications of a tool in comparison to chance, while Markedness combines precision and its inverse metric to measure how marked the classifications of a recommender are in comparison to chance.

Recall	$\frac{TP}{TP+FN}$
	Proportion of positive cases that are correctly classified as positive. Also called true positive rate or sensitivity.
Precision	$\frac{TP}{TP+FP}$
	Proportion of the classified positive cases that are correctly classified. Also referred to as true positive accuracy, positive predictive value or confidence.
F-measure	$2 \times \frac{prec \times recall}{prec+recall} = \frac{2 \times TP}{2 \times TP+FN+FP}$
	Represents the harmonic mean of precision and recall.
Informedness	$\frac{TP}{TP+FN} - \frac{FP}{FP+TN}$
	Quantifies how consistently the predictor predicts the outcome, i.e. how informed a predictor is for the specified condition, versus chance.
Markedness	$\frac{TP}{TP+FP} - \frac{FN}{FN+TN}$
	Quantifies how consistently the outcome has the predictor as a marker, i.e. how marked a condition is for the specified predictor, versus chance.

Table 8.1: Selection of Metrics for Benchmarking Security Tools ([10]).
Note: *TP, TN, FP* and *FN* stands for *True Positives, True Negatives, False Positives* and *False Negatives*, respectively.

An important fact identified in previous research is that, experts declared different levels of familiarity with the use of each proposed metric. Despite this, they only rely on the use of one metric which provides no opportunity for such experts to take part in the analysis process. As a result, such familiarity, although important, cannot be taken

in consideration and influence the relative importance provided to metrics. This is a challenge that in practice, means that not all experts opinions should be considered as equally pertinent, but their relevance must be modulated according to their familiarity with the metrics under analysis. Our proposal will address this issue.

8.2.2 Consideration of Analysis Scenarios

If it is true that the usefulness of a benchmark depends directly on the set of selected metrics, it is also true that it also depends on the adequacy of those metrics to the specific scenario in which the benchmark is being executed [83]. From this perspective the definition of the benchmarking scenario is key for the selection of the metrics that provide the relevant insight according to the priorities of the benchmark user.

Scenario	Requirements
Business-critical	The *Business-critical* scenario represents systems with high-demanding security requirements, where the exploitation of a vulnerability can be reflected in economical or reputation losses. These are systems such as home banking, stock trading, or large-scale e-commerce. The development of this kind of systems is assumed to have enough resources to deal with all reported vulnerabilities, even if they are wrongly classified (false positives). Thus, the goal is to select a tool able to detect the highest number of vulnerabilities, leaving undetected the lowest number possible.
Heightened-critical	The *Heightened-critical* scenario represents those systems where the applications are subjected to high security requirements (but not as those running in business-critical scenarios). This could be the case of applications dealing with sensitive data, like governmental portals or large scale social networks. Here, the aim is to detect the highest number of vulnerabilities possible, but unlike business-critical, it cannot be done at any cost, so it is necessary to avoid tools reporting too many false positives.
Best effort	The *Best effort* scenario represents applications that are less exposed to attacks or are not as important as the previous scenarios. Time or budgets constraints in this scenario have to be considered, as the resources available to fix reported potential vulnerabilities are limited, thus a trade off must be found. Development of big web portals where attacks represent small direct financial loss or intranet applications that are less exposed to external attacks, are some examples. Here, the goal is to look for tools able to detect a high number of vulnerabilities while reporting low number of false positives.
Minimum effort	The *Minimum effort* scenario represents low resources applications with not much criticality concerns that might not be subjected to a lot of external attacks. Due to budget reasons, the time and money available to fix vulnerabilities are usually tight. Hence, tools reporting the lowest number of false positives with high confidence in the reported vulnerabilities are desired for this scenario. This would be the case of development of CMSs for small and medium companies, and information/advertising web sites.

Table 8.2: Scenarios for the use/analysis of Security Tools ([10]).

Considering the criticality that an exploited vulnerability would have on the system, and the needs and resources of the users of vulnerability detection tools, four common benchmarking scenarios in the field of binary detection of vulnerabilities have been identified in [10]: i) Business-critical, ii) Heightened-critical, iii) Best effort, and iv) Minimum effort. In the *business-critical scenario*, the best security tools are those able to detect the highest number of vulnerabilities, while leaving undetected the lowest number possible. In the *heightened critical scenario*, detecting the highest number of vulnerabilities is also important, but it cannot be done at any cost, so it is necessary to avoid tools reporting too many false positives. Then, the *best effort scenario* looks for tools able to provide a good

balance between high-level detection and false positives. Finally, in the *minimum effort scenario*, the goal is to look for tools reporting the lowest number of false positives with high confidence in the reported vulnerabilitites. Further information on these scenarios, including examples of the types of systems included in each one, is provided in Table 8.2.

It is worth mentioning that, even in [10], where several metrics were proposed for benchmarking vulnerability detection tools, only one of such metrics was considered for scoring and ranking purposes per scenario, although a second metric was also nominated as tiebreaker. These recommendations are reported in Table 8.3.

Scenario	Main Metric	Tiebreaker
Business-critical	Recall	Precision
Heightened-critical	Informedness	Recall
Best effort	F-measure	Recall
Minimum effort	Markedness	Precision

Table 8.3: Recommended metrics (from [10]).

8.2.3 Do We Really Need a New Analysis Approach?

If one accepts that i) security tools are currently ranked and selected using only part of the all the available information issued from the benchmarking process, and ii) the advice of experts is not required when analyzing such information, then the answer to the question may be *no*.

However, the acceptance of these assumptions attempts against the principles of accuracy and confidence that everyone assumes when using the results of a benchmark. In our case we are dealing with a multi-criteria evaluation problem (selection of the best alternative using multiple benchmark metrics) requiring not only context-awareness (consideration of the influence of benchmarking scenarios in the analysis process) but also domain expert support (due to the technical characteristics and the level of criticality of the targeted tools). To the best of our knowledge, no analysis approach has been proposed so far in order to address all these requirements when benchmarking security tools.

8.3 A Multi-criteria Analysis Approach With Expert Support

MABRES is the acronym of *Multi-criteria Analysis of Benchmark Results with Expert Support*, and is the name of the analysis methodology that we propose. This section opens with a high-level description of MABRES. Then, a subsection will be devoted to describe each phase of the methodology.

8.3.1 Approach overview

As Figure 8.1 shows, MARBRES complements the traditional benchmarking process. The benchmark targets (in our case, the tools under benchmarking) are instantiated attending to an operational profile that includes, among other things, a workload and an attack/vulnerability-load. Traditionally, the measurements retrieved from experimentation are carefully treated in order to deduce the metrics that are finally reported.

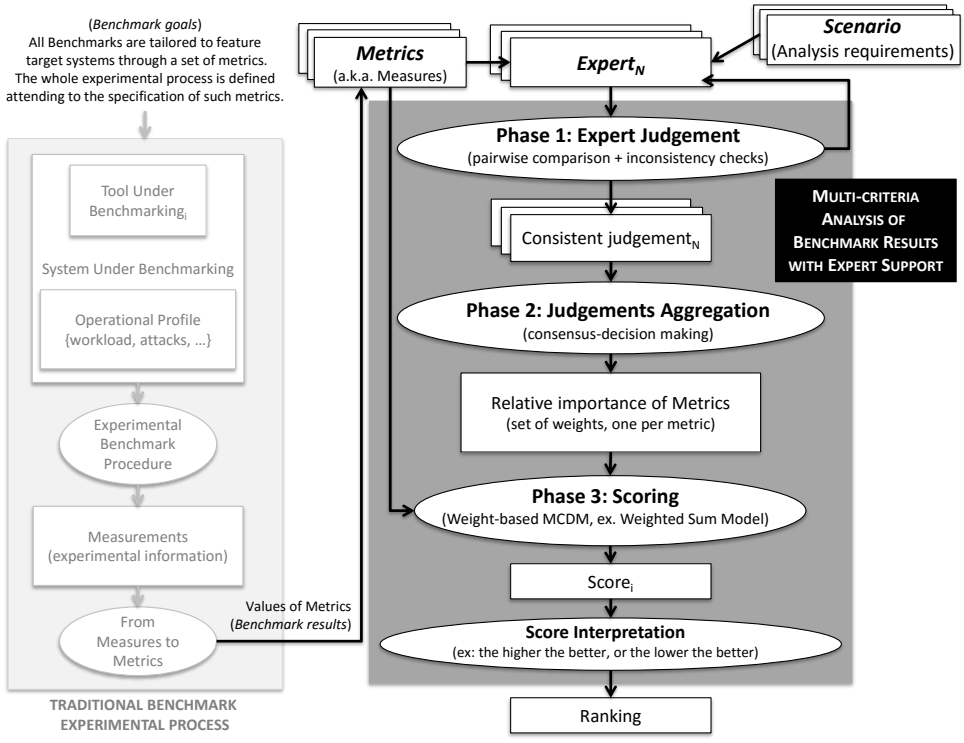

Figure 8.1: The MABRES Approach: a Multi-criteria Analysis of Benchmark Results with Expert Support

The analysis process is typically left unspecified in benchmarks. So, benchmark users take the responsibility of carrying out such analysis properly. MARBRES wants to place a certain order in the analysis of benchmark results by promoting a systematic process that keeps the solution as simple as possible to ease its use, understanding, and final explanation. At the same time, it also guarantees the traceability from metrics to scores and vice-versa during the whole analysis process.

The analysis approach supported by MARBRES is detailed in Figure 8.1 and it runs just after the traditional benchmark process. It relies on the existence of i) a set of metrics featuring each targeted security tool (multi-criteria component), and ii) the specification of one or several benchmarking scenarios (context-awareness component). As exemplified in the next section (case study), the set of tools under study are analyzed from the perspective of the five metrics (see Table 8.3) and considering the analysis requirements imposed by the four benchmarking scenarios identified in Table 8.2.

MARBRES also enables the involvement of one or several domain experts (expert support component) in the analysis process. As already pointed out in the introduction of this paper, even when no expert is explicitly involved in the analysis process, it can be assumed that the benchmark user implicitly becomes the expert when participating in the ranking of the evaluated alternatives.

The methodology proposed in MARBRES works in three successive phases:

1. First, experts compare available metrics in pairs attending to the analysis requirements imposed by each considered scenario. This is the *Expert Judgement* phase, which results in a pairwise comparison matrix per expert and scenario. Each one of these matrixes is then automatically processed to detect inconsistencies in carried out comparisons. As a result, inconsistent comparisons can be reviewed or discarded, while consistent ones can be finally processed. The final output of this step is a set of vectors capturing the judgments provided by experts.

2. The second phase, the so-called *Judgements Aggregation* phase, looks for establishing a consensus among all individual judgements provided by experts. This consensus is expressed as a single vector of weights reflecting the relative importance that is globally provided by experts to metrics in each scenario. All this process is carried out automatically.

3. Once the importance of each metric is set per scenario, the third phase, named the *Scoring* phase, relies on the use of multi-criteria decision making analysis (MCDM) techniques to compute a final score for each benchmarked alternative. Since we are working with weights, any weight-based MCDM can be used in our case. In order to keep things as simple as possible, our recommendation is to use of the widely used and well-known Weighted Sum Model method. With this model,

each benchmark result (metric) is treated as a different selection criterion and its influence in the final resulting score is pondered attending to the weight that the consensus among experts has attributed to it.

The final result of these three phases is thus a single score that must be subsequently interpreted following a pre-established criteria, such as the-higher-the-better or the-lower-the-better. This is how the evaluated alternative is integrated in the final produced ranking.

8.3.2 Phase 1: Individual Expert Judgement

In this initial phase, experts judge the importance of each particular metric over another. It is obvious that this importance is highly determined by the considered scenario. For example, let us consider two metrics in the field of networking such as *throughput* (amount of information transmitted per second) and *integrity* (percentage of packets received with its information intact). In an scenario where a network is conformed by users exchanging large files with public data, *throughput* might be more important than *integrity*, since the fast exchange of information would be more relevant, and corrupted packets could be requested again. On the other hand, in a network where users exchange small files containing private data, the *integrity* of the packets would be more important than having a high *throughput*. This way, an expert needs to have a good insight of the *specification of the scenario* to perform an informed decision on which metrics are more important than others, and to what extent.

When it comes to compare multiple metrics simultaneously, human subjectiveness makes it difficult for a person to be accurate when considering more than 2 elements at the same time. However, the comparison of pairs is something that humans can easily do. Indeed, it is well-known than paired comparison is less error-prone than considering all metrics at the same time, and it can be easily (re)checked in case of finding any inconsistency among the comparisions carried out.

The pairwise comparison method enables weighting the relative importance of metrics, while allowing experts to use quantitative values to express qualitative decisions. It is part of the Analytic Hierarchy Process (AHP) [89], a famous decision-making framework developed by mathematicians in the 80s and used today in many different application domains, ranging from business to engineering [24, 41, 101].

In practice, experts perform a pairwise comparison of the metrics using a 1 to 9 scale known as the *Fundamental Scale of Absolute Numbers for Pairwise Comparison* (see Table 8.4) to translate their qualitative decisions into quantitative values. With the assistance of this scale, the experts compare the metrics two-by-two and their answers are used to fill a *pairwise comparison matrix* from which the requirements are calculated.

Definition	Description	Intensity[a]
Equal	A and B are equally important	1
Moderate	A is somewhat more important than B	3
Strong	A is much more important than B	5
Very strong	A is very much more important than B	7
Extreme	A is absolutely more important than B	9

[a] Intensities of 2, 4, 6 and 8 can be used to express intermediate values. Very close importance values can be represented with 1.1–1.9.

Table 8.4: The fundamental scale of absolute numbers for pairwise comparison

The comparison of L metrics leads to the definition of a $L \times L$ matrix, as shown in Equation 8.1. From every comparison between two metrics, the expert determines which metric is more important than the other, and quantify that importance using the intensities shown in Table 8.4. Since the intensity of a metric M_i with respect to another metric M_j is represented by x_{ij}, the opposite intensity is $x_{ji} = 1/x_{ij}$, which makes the matrix reciprocal. Hence, $\forall i, j \in L : x_{ij} \times x_{ji} = 1$.

$$\begin{array}{c} \\ M_1 \\ M_2 \\ \vdots \\ M_L \end{array} \begin{array}{c} \begin{array}{cccc} M_1 & M_2 & \cdots & M_L \end{array} \\ \begin{pmatrix} 1 & x_{12} & \cdots & x_{1L} \\ x_{21} & 1 & \cdots & x_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ x_{L1} & x_{L2} & \cdots & 1 \end{pmatrix} \end{array} \tag{8.1}$$

Figure 8.2 provides a concrete example to explain how a real pairwise comparison matrix looks like. In this example 3 different metrics A, B and C are compared. The expert considers that B is moderately more important than A, B is very much important than C, and A is much more important than C. Since the matrix is reciprocal, the matrix can be filled with these 3 comparisons.

Once the pairwise comparison matrix is available, a *priority vector* is computed. This vector has as many elements as considered metrics, each element weighting the relative importance provided by the expert to the related metric. There are two main prioritization methods that can be used to compute the priority vector: the *eigenvalue* method [90] and the *row geometric mean* (RGM) method [29]. The work done in [31, 59] shows that the difference in the output from the application of any of these methods is meaningless, although the RGM method requires in general less computational power. This is why in this paper we propose the use of the RGM method.

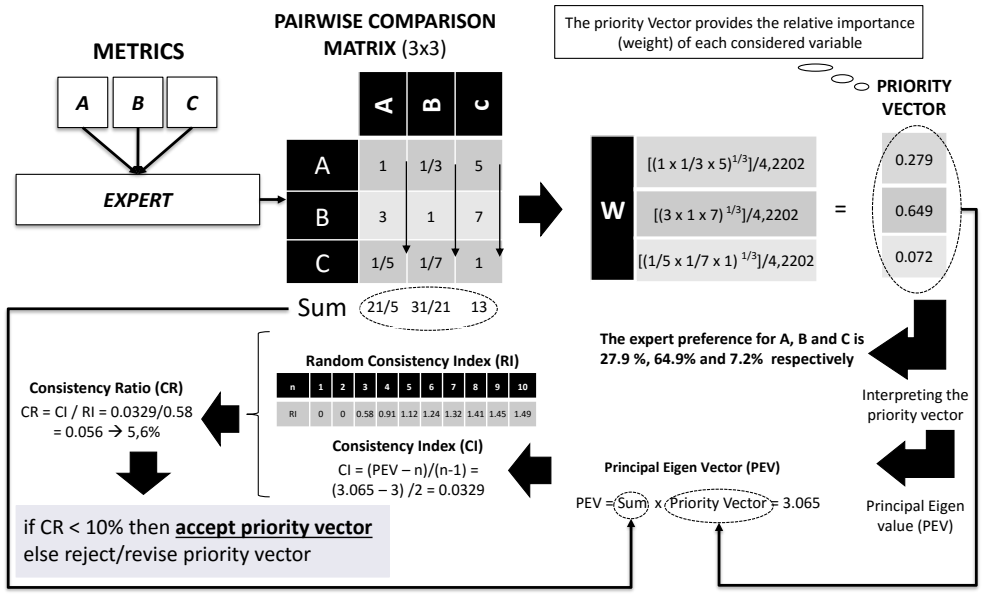

Figure 8.2: Capture and Automatic Processing of the Individual Judgement of an Expert

The procedure followed by the RGM method to compute the priority vector is depicted in Equation 8.2 and it follows three successive steps: *i)* compute the geometric mean for each row of the pairwise comparison matrix, *ii)* sum up all computed geometric means, and *iii)* divide each geometric mean by the resulting sum. The result is a priority vector $w = (w_1, \cdots, w_L)$ containing L different weights (one per metric) that should respect that $w_j \geq 0$ and $\sum_{j=1}^{L} w_j = 1$.

$$w_i = \frac{\sqrt[1/L]{\prod_{j=1}^{L} x_{ij}}}{\sum_{i=1}^{L} \left(\sqrt[1/L]{\prod_{j=1}^{L} x_{ij}} \right)} \tag{8.2}$$

This process is illustrated in Figure 8.2. The figure also shows how the various weights contained in the resulting priority vector can be interpreted. As can be seen, the values contained in the priority vector represent the relative importance declared by the expert to each metric.

The main problem with the use or pairwise comparison matrices is that humans are involved in their definition and, consequently, such matrices may contain inconsistencies

due to (subjective) interpretation. In [75], it was proposed a statistically reliable estimate to quantify the consistency of the resulting priority vector, the so-called consistency ratio (CR). As Figure 8.2 shows, the CR is computed in three successive steps. First the Priority Eigen Vector (PEV) is calculated by multiplying the sum of the various columns of the pairwise comparison matrix ($1xL$ matrix) and the weights contained in the priority vector ($Lx1$ matrix). Then, a consistency index (CI) is deduced attending to the PEV and the number of metrics under study (L in our case). Finally, the consistency ratio (CR) can be obtained by normalizing the CI to the random consistency index (RI) that is directly obtained from a table defined in [88]. The CR checks that intensities representing the relative importance between elements of the matrix are consistent: a matrix with a $CR < 0.1$ is considered consistent. Inconsistent matrices can be either neglected or reviewed until they become consistent. Figure 8.2 shows this process in action. The finally computed value for CR of 0.056 is smaller than 0.1, so the priority vector is accepted.

The formal justification of the afore-described process falls beyond our purpose, although interested readers may refer to [3] and [88] for further details. The important thing is that this process is representative for the community, since it is largely adopted by the academia and the industry [90], and it can be fully automated, which eases its use.

8.3.3 Phase 2: Aggregation of Individual Judgments

At this point in the methodology, N experts have determined the relative importance among the L metrics considering the scenario of application for the target benchmark tools, thus providing a set of N requirements (one per expert).

The aggregation of individual judgments can be carried out in multiple ways using, for instance, consensus-decision making methods or voting theories. Nevertheless, it must be clear from the very beginning that we are not looking for a winner or a loser. The goal is to reach an agreement that accounts for the individual contributions of all experts. However, this does not mean that all the judgments will be treated equally. As already mentioned in Section II, the relevance of each judgment will be determined attending to the level of familiarity of each expert with the considered metrics and/or analysis scenarios.

There are two main methods that have proven to be useful in group decision making when considering decisions expressed as priority vectors: the *aggregation of individual judgments* (AIJ) and the *aggregation of individual priorities* (AIP) [47]. Despite their differences, when the RGMM method is used to calculate the priority vector (see previous subsection), both methods are equivalent, thus leading to the same set of group priorities [14, 32], i.e. the same consensus.

At the light of this situation, our decision has been to use the AIJ method. The main reason is that AIJ promotes a judgment aggregation approach reusing existing pairwise comparison matrices and producing a new type of pairwise comparison matrix, called the group comparison matrix (GCM). Every element of a GCM is the result of computing the weighted geometric mean of the elements located at the very same position in all the pairwise comparison matrices provided by experts. For example, the position (2,1) (second row, first column) of the GCM is calculated as the weighted geometric mean of all the values in the position (2,1) of all the experts' individual pairwise comparison matrices. Obviously, it is assumed that all expert comparison matrices under consideration are consistent.

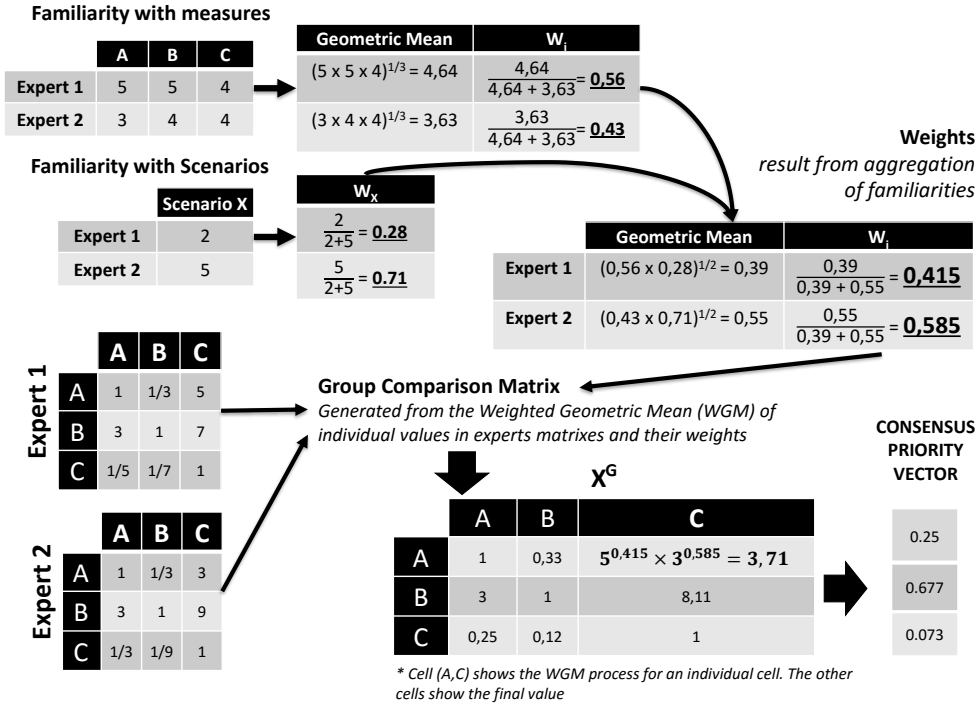

Figure 8.3: Weighting the contribution of experts to the final aggregation of opinions

The weights that are required must be settled attending to the familiarity of each expert with each metric and analysis scenario. The proposal is to directly ask experts about such familiarity and then aggregate provided answers using the geometric mean (GM). In this case the approach must be deployed recursively as shown in Figure 8.3. First, the familiarity reported by experts with measures and scenarios is managed separately. In each case, the GM of the provided answers is computed for each expert, and the result

is normalized with the sum of all calculated GMs. As it is shown in Figure 8.3, since the familiarity of an expert with an scenario is determined by a single value, the GM of that value, is the value itself, thus only the normalization by the sum of all values is required. The weight for an expert in a given scenario is then calculated by applying the GM of both his weights (the one from the measures and the one for the scenario), which provides the weight assigned to such expert. When calculated for all experts, this results are a percentage that reflects the relative importance among experts opinions to contribute to the *group consensus matrix*. The advantage of this proposal is that it scales up with the number of considered metrics, scenarios and experts, and at the same time it let us adjust experts weights according to their level of expertise in the proposed scenarios.

Once the contribution of experts to the final aggregation of judgment is determined, the group comparison matrix X^G can be computed as described in Equation 8.3. Here it is considered a set of N experts, denoted by $E = \{e_1, \cdots, e_N\}$, where every expert has an associated weight that determines its contribution to the X^G matrix. These weights are denoted by $\omega = \{\omega_1, \cdots, \omega_N\}$, where $\omega_i \geq 0$ and $\sum_{i=1}^{N} \omega_i = 1$. Hence, the X^G matrix is built considering all the experts opinion, where the pairwise comparison matrix defined by an expert e_k is represented as X^k, and its value in the position (i, j) as x_{ij}^k. So, the value x_{ij}^G of matrix X^G is the result of applying the weighted geometric mean to the element x_{ij} of all the matrices defined by the experts. A simple example is shown in Figure 8.3, where the X^G is calculated from the pairwise comparison matrices of two experts.

$$x_{ij}^G = \prod_{k=1}^{N} \left(x_{ij}^k\right)^{\omega^k} \tag{8.3}$$

As can be easily understood, X^G, the group comparison matrix, is in essence a pairwise comparison matrix. So, the same procedure defined for the analysis of this type of matrices (see Phase 1 of the proposal) can be applied in this case to deduce the priority vector associated to X^G and to reason about the consistency of such vector (see Figure 8.3). If X^G is consistent, then the resulting priority vector can be accepted. For the sake of distinction, let's call this vector as the *consensus priority vector*. It will have as many elements as metrics under consideration, L in our case, and it will be denoted as $wc = \{wc_1, \cdots, wc_{cL}\}$, where $wc_i \geq 0$ and $\sum_{i=1}^{L} wc_i = 1$.

8.3.4 Phase 3: Scoring

The inputs to this phase are the *consensus priority vector* previously computed and the set of values obtained by each alternative under evaluation for the L metrics under study. The result is the final score that will be later interpreted in order to establish a final ranking among all benchmark targets.

This phase is maybe the one that can lead to more controversy in the whole approach, since one of the most challenging issues when aggregating metrics in benchmarking is to properly capture in a single score information of the system or tool under assessment[79]. The goal is not only to compute a score, but rather to use the most simple, easy to use, understandable and explainable method. At the same time, the scoring process must be traceable in order to clearly identified how metrics are transformed into scores and what is the contribution of the various considered metrics. These are mandatory requirements to keep the analysis approach sound and representative to the community of potential benchmark users.

Although several methodologies are available to cope with the above requirements, they can always be criticized for one drawback or another. As mentioned in [4], the mathematical addition, for instance, cannot be directly applied to all metrics; the central tendency methods often hide underlying distributions; wealth inequality and distribution fitting techniques are hard to interpret and their results are usually difficult to trace back to the originally considered metrics; and finally, custom formulae are hard to validate.

We propose to score benchmarked alternatives using the Weighted Sum Model. The selection of this method is not casual:

- First, it is the best known and simplest multi-criteria decision method (MCDM) approach for comparing and ranking a number of tools in terms of a complex set of decision criteria (metrics and their relative importance).
- Second, it adapts well to the type of metrics under consideration since i) all of them can be expressed in the same unit, and ii) all of them can be interpreted following a same benefit criteria, i.e. the higher the values are, the better it is. It is worth mentioning that, although all metrics are expressed as values between 0 and 1, the informedness and the markedness metrics are expressed as values between -1 and 1. However this problem can be easily solved by normalizing these two metrics between 0 and 1 using the formulas provided in Table 8.5.
- Third, the computation of scores can be directly carried out using the set of inputs already available at this phase of the proposal.

- Fourth, the interpretation of resulting scores is very easy and also follows a benefit criteria.
- And last, but not least, it has been used for years in a way or another by benchmark users, so its use in meaningful for everyone in the domain of benchmarking.

Metric	Formula
Informedness	$\frac{\frac{TP}{TP+FN} - \frac{FP}{FP+TN} + 1}{2}$
Markedness	$\frac{\frac{TP}{TP+FP} - \frac{FN}{FN+TN} + 1}{2}$

Table 8.5: Normalization applied to the Informedness and Markedness metrics for the purpose of scoring

The mathematical notation of the scoring process for an alternative A_i in an scenario S_x is shown in Equation 8.4. Here, wc_k denotes the priority calculated in the consensus priority vector (computed in the previous phase) for the kth metric. In the other hand, m_k refers to the value that alternative A_i has obtained in the kth metric. Thus the final score for an alternative is the addition of multiplying the value obtained in each metric by the weight calculated for that metric by the experts (Weighted Sum Model).

$$Score(A_i, S_x) = \sum_{k=1}^{L} (m_k \times wc_k) \tag{8.4}$$

Following this process, one score is finally attributed to each alternative in each one of the considered analysis scenarios. As a result, it must be repeated as many times as $alternatives \times scenarios$ are considered.

8.4 Case Studies

This section reports the analysis of two different sets of benchmark metrics. The former set corresponds to the benchmarking of 10 different vulnerability detection tools. The latter focuses on the benchmarking of 11 different intrusion detection system (IDS) tools for SQL injection attacks in web applications. The main difference between these data sets is that the first one can be nearly analyzed at a first sight, while the second presents a high variability of winners and losers depending on the considered metric, so its analysis is far less evident. The first case study will test the ability of the proposal to rank evaluated alternatives in agreement to what it is dictated by the common sense. The second case study will study the effect on rankings of neglecting part of the information pro-

vided by the available metrics, and the contrary effect of considering the contribution of all metrics and the advice of experts.

In both case studies, the use of MARBRES will be carried out attending to the requirements defined for the four scenarios listed in Table 8.2 and the metrics under analysis will be the ones computed according to Table 8.1. Remember that the formulas for *Informedness* and *Markedness* are adapted in MARBRES for the purpose of making them compatible with the rest during the scoring phase, as already explained in Section 8.3.4.

8.4.1 Capturing expert's priorities

In order to compare different tools, it is necessary to determine the *consensus priority vector* for each scenario. To perform these case studies, a group of researchers with expertise in dealing with metrics driven from binary classifications (TP, TN, FP and FN) and benchmarking processes, were invited to participate in order to determine the *consensus priority vectors*. From all the invitations, 21 researchers accepted to contribute to our research.

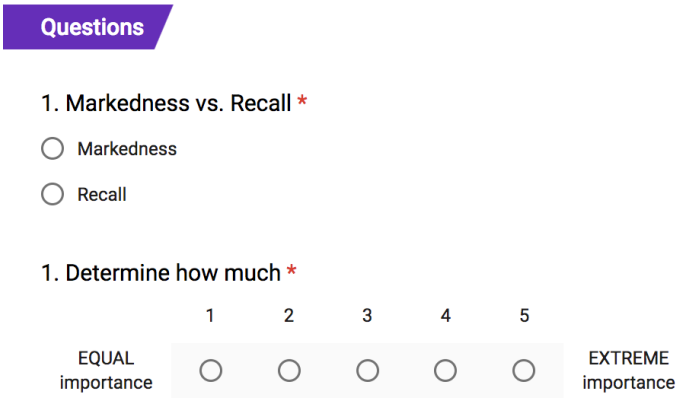

Figure 8.4: Example of the questions formulated to experts

The experts were asked to carry out a paired comparison of all the considered metrics for each one of the four considered analysis scenarios. An online questionnaire[1] was prepared for that purpose. Each pair of metrics is compared for every scenario through two questions that determine the relative importance between them according to the expert. The first question asks for the selection of the preferred metric, while the second quantifies (following the scale presented in Table 8.4) the expert's opinion on how much

[1] Available at https://goo.gl/forms/EEmkUmLIj20nMJS33

important the preferred metric is with respect to the other. An example of these questions is provided in Figure 8.4.

It is important to say that some experts participating in the comparison of metrics experienced certain problems when answering the questionnaire. As a result, certain pairwise comparison matrices were inconsistent, so they required some revision. This was something expected since, despite the simplicity of the approach, a certain training is required to properly make all the comparisons. However, experts did not have such training by the time they filled the proposed questionnaire. Finally, a total of 27 opinions, out of 84 provided (21 experts evaluated the metrics from the perspective of 4 different analysis scenarios), i.e. 32%, were finally left out of our study.

The familiarity of the experts with the metrics and scenarios was captured by the questionnaire. A value between 1 and 5 was given to determine such familiarity, being 1 the lowest level of familiarity and 5 the highest. With the reported information, the set of group comparison matrices (one per scenario) and their respective consensus priority vectors, were deduced. All the group comparison matrices passed the consistency check, so all the consensus priority vectors were pertinent for the analysis of metrics. Resulting vectors are listed in Table 8.6. The table also reports the set of (consistent) pairwise comparisons (expert opinions) taken into consideration per scenario.

		Metric				
Scenario	**#E**	*Rec.*	*Prec.*	*F-Meas.*	*Inform.*	*Mark.*
BC	14	0.58	0.08	0.12	0.14	0.08
HC	13	0.32	0.16	0.19	0.22	0.11
BE	17	0.11	0.16	0.30	0.22	0.21
ME	13	0.07	0.34	0.12	0.20	0.27

Table 8.6: Consensus Priority Vectors for the Considered Scenarios

It can be seen in Table 8.6 that the number of experts involved in each scenario varies. This is due to the aforementioned situation that some of the experts had difficulties when performing the pairwise comparison of the metrics, which lead them to provide inconsistent matrices that could not be used in the analysis. Although the higher number of consistent matrices we have, the better, it is true though that involving a high number of experts in this type of analysis is not an easy task.

8.4.2 Case Study 1: Ranking Vulnerability Detection Tools

The data set analyzed in this subsection is the one originally introduced and analyzed in [10] using a single metric and a tiebreaker per scenario, as already reported in Table 8.3. It is important to clarify from the very beginning that the goal here is not to corroborate or redo what other authors have already published, but rather to compare their results with the ones provided by our analysis proposal.

Table 8.7 lists the set of metrics under analysis. Such metrics results from the benchmarking of the 10 different vulnerability detection tools. It is to underline that the names of the evaluated tools is irrelevant for the purposes of this paper. So, they are named using anonymous labels of type TXX, where "XX" takes a value from 01 to 10.

	Metric				
Tool	*Recall*	*Precision*	*FMeasure*	*Informedness*	*Markedness*
T01	0.793	1.000	0.885	0.793	0.953
T02	0.552	0.923	0.691	0.541	0.825
T03	1.000	0.640	0.780	0.864	0.640
T04	0.149	0.325	0.205	0.075	0.144
T05	0.753	1.000	0.859	0.753	0.903
T06	0.323	0.455	0.378	0.156	0.195
T07	0.241	0.388	0.297	0.076	0.105
T08	0.019	1.000	0.037	0.019	0.702
T09	0.241	0.567	0.338	0.161	0.304
T10	0.741	1.000	0.851	0.741	0.899

Table 8.7: Benchmark Result under Analysis in Case Study 1 ([10]).

From the consensus priority vectors calculated for each scenario (see Table 8.6) and the results obtained from the evaluation of each tool (see Table 8.7), the scoring and ranking of the security tools was performed. Table 8.8 shows these rankings and compare them with the ones that would have been produced, if the recommendations provided in Table 8.3 had been followed.

An eye-based analysis of the metrics under consideration (see Table 8.7) shows that T03 outperforms all the rest when Recall and Informedness are considered, while in these cases T01, T05 and T10 go to the second place. The reverse happens when taking into account the other three metrics, i.e. Precision, F-Measure and Markedness. In that case, T01, T05 and T10 provide similar scores and they outperform the rest, while T03 goes to the second place. If all the measures are considered together, it seems quite intuitive that the best alternative should be either T03, T01, T05 or T10, and in any case, these four alternatives will be always arranged among the four best alternatives in all the scenarios.

Business-Critical		
SM	MABRES	Score
T03	T03	0.932
T01	T01	0.846
T05	T05	0.814
T10	T10	0.804
T02	T02	0.651
T06	T06	0.401
T07	T09 (↑ 1)	0.359
T09	T07 (↓ 1)	0.330
T04	T04	0.264
T08	T08	0.220

Heightened-Critical		
SM	MABRES	Score
T03	T01 (↑ 1)	0.882
T01	T03 (↓ 1)	0.881
T05	T05	0.856
T10	T10	0.849
T02	T02	0.719
T09	T06 (↑ 1)	0.448
T06	T09 (↓ 1)	0.434
T07	T07	0.383
T04	T08 (↑ 1)	0.364
T08	T04 (↓ 1)	0.332

Best Effort		
SM	MABRES	Score
T01	T01	0.916
T05	T05	0.894
T10	T10	0.889
T03	T03	0.824
T02	T02	0.780
T06	T09 (↑ 1)	0.487
T09	T06 (↓ 1)	0.478
T07	T08 (↑ 2)	0.470
T04	T07 (↓ 1)	0.416
T08	T04 (↓ 1)	0.374

Minimum Effort		
SM	MABRES	Score
T01	T01	0.951
T05	T05	0.935
T10	T10	0.931
T02	T02	0.848
T08	T03 (↑ 1)	0.782
T03	T08 (↓ 1)	0.711
T09	T09	0.552
T06	T06	0.504
T04	T07 (↑ 1)	0.446
T07	T04 (↓ 1)	0.415

Table 8.8: Rankings generated with MABRES Scores vs. Rankings obtained using a Single Metric (SM) for Case Study 1

This is what dictates the common sense and what can be inferred from results without using any sophisticated analysis methodology.

The use of MABRES and the Single Metric (SM) approaches corroborates this "common sense analysis", hence, as it can be seen, the four mentioned alternatives are the most relevant ones in all the proposed rankings, except in the case of the Minimum effort scenario where T02 goes to the fourth place of the ranking.

Although not very exciting at a first sight, these results support the opinion that MABRES do not perturb the expected rankings despite the simultaneous consideration of all the available metrics and the opinion of 21 experts. From that viewpoint, this case study ver-

ifies the pertinence of the resulting rankings and illustrates the feasibility of the approach with a simple data set.

One can then be tempted to infer from this situation that using such a sophisticated analysis approach is useless, since there exists a simpler one, the SM, leading to the same conclusions. This is unfortunately false. The results seem to agree with this assessment only if the metrics under consideration present enough variability among them to clearly identify winners and losers in each considered analysis scenario. But, what if that eye-based classification is not so evident? This situation is studied in the next section, through a new data set obtained from the evaluation of 11 different intrusion detection system (IDS) tools in web applications.

8.4.3 Case Study 2: Ranking IDS Tools for Web applications

The evaluation of Intrusion Detection System (IDS) tools for SQL injection attacks in web applications can be also addressed using MABRES. With an SQL injection the attacker may read, alter or destroy the content of a database and these IDS tools are used to detect when such attacks are occurring. Their detection rates are also described in terms of *true positives*, *true negatives*, *false positives* and *false negatives*, then the same metrics and the same scenarios can be used to compare and rank this second type of tools.

Tool	Recall	Precision	FMeasure	Informedness	Markedness
T01	0.790	0.876	0.831	0.275	0.210
T02	0.350	0.862	0.498	0.091	0.059
T03	0.275	0.993	0.431	0.266	0.221
T04	0.193	1	0.324	0.193	0.211
T05	0.089	1	0.163	0.089	0.192
T06	0.538	1	0.700	0.538	0.795
T07	0.162	0.966	0.277	0.135	0.167
T08	0.180	0.940	0.302	0.127	0.140
T09	0.794	0.615	0.693	0.517	0.478
T10	0.882	0.493	0.633	0.377	0.376
T11	0.293	1	0.453	0.293	0.327

Table 8.9: Metrics under Analysis in Case Study 2

Since the set of considered metrics and the analysis scenarios are the same, the pairwise comparison matrices, the group comparison matrices, and more importantly, the final set of consensus vector priorities (shown in Table 8.6) are the same. So basically, the same analysis previously carried out is now applied on a different data set. This will enable

us to study the effect of neglecting the part of the information provided by unconsidered metrics during the analysis. And, on the other hand, how rankings change when this information and the advice of experts is taken into consideration.

Table 8.9 shows the metrics obtained from the benchmarking of a total of eleven different IDS tools ($T01, T02, \cdots, T11$). Their analysis using a single metric (SM) and the MABRES approach are depicted in Table 8.10.

Business-Critical		
SM	MABRES	Score
T10	T10	0.786
T09	T09	0.764
T01	T01	0.759
T06	T06	0.647
T02	T02	0.441
T11	T11	0.437
T03	T03	0.418
T04	T04	0.352
T08	T08	0.33
T07	T07	0.32
T05	T05	0.262

Heightened-Critical		
SM	MABRES	Score
T06	T01 ($\uparrow$ 4)	0.747
T09	T09	0.739
T10	T06 ($\downarrow$ 2)	0.724
T11	T10 ($\downarrow$ 1)	0.717
T01	T11 ($\downarrow$ 1)	0.545
T03	T03	0.525
T04	T02 ($\uparrow$ 3)	0.512
T07	T04 ($\downarrow$ 1)	0.47
T08	T08	0.442
T02	T07 ($\downarrow$ 2)	0.438
T05	T05	0.393

Best Effort		
SM	MABRES	Score
T01	T06 ($\uparrow$ 1)	0.789
T06	T01 ($\downarrow$ 1)	0.739
T09	T09	0.717
T10	T10	0.663
T02	T11 ($\uparrow$ 1)	0.608
T11	T03 ($\uparrow$ 1)	0.583
T03	T02 ($\downarrow$ 2)	0.554
T04	T04	0.535
T08	T08	0.503
T07	T07	0.501
T05	T05	0.462

Minimum Effort		
SM	MABRES	Score
T06	T06	0.87
T09	T01 ($\uparrow$ 5)	0.74
T10	T11 ($\uparrow$ 1)	0.732
T11	T03 ($\uparrow$ 1)	0.707
T03	T09 ($\downarrow$ 3)	0.698
T04	T04	0.686
T01	T07 ($\uparrow$ 2)	0.655
T05	T05	0.651
T07	T08 ($\uparrow$ 1)	0.645
T08	T02 ($\uparrow$ 1)	0.633
T02	T10 ($\downarrow$ 8)	0.626

Table 8.10: Rankings generated with MABRES Scores vs. Rankings obtained using a Single Metric (SM) for Case Study 2

A first eye-based analysis of such metrics shows that T10 outperforms the rest when considering *Recall*, although T01 and T09 are also doing quite well for the same metric.

However, T10 does not stand out for any other metric. As far as Precision is concerned, T03, T07 and T08 are nearly equally good, with a slight advantage for T07, but negligible. The score of T01 and T02 is also good for that metric. For *F-measure*, T01 is the best alternative, although T06 and T09 are quite near. These tools (T06 and T09) are however the ones obtaining the more interesting *Informedness* value (despite being the best, they are still quite low). Finally, tool T06 is clearly a very good choice when analysis the selection problem from the perspective of the *Markedness* metric. The point there is not that T06 provides an impressive *Markedness* value, but rather that the rest of tools, except maybe T09, provide a rather low value.

Comparing with the previous case study, selecting the best alternative is now much more difficult. On the one hand, there is no tool appearing in all the rankings in a good position. So, applying the principle of selecting the tool that is good in average is not an option. On the other hand, the variety of classifications obtained attending to each particular metric is so broad, that no groups of tools can be identified.

The rankings reported by Table 8.10 show that, although in some cases using a Single Metric (SM) can be as acceptable as using the MABRES approach, in other scenarios the difference between SM-based and MABRES-based rankings is very important. In the *Business-Critical* scenario both rankings are identical. However, differences appear when the considered scores are taken into consideration. We should remember that the *Recall* metric is the more important one for the *Business-Critical* scenario. Attending to the results in that metric, T10 is 7.8% better than T09, and the difference between T09 and T01 is negligible (less than 1%). However, the difference between the MABRES scores provided to the first (T10), second (T09) and third (T01) alternative are not really so important (less than 3% among all three). As a result, the clear winner for SM (T10) is not so winner for MABRES, which considers all the three alternatives (T10, T09 and T01) as (nearly) equally good for the *Business-Critical* scenario. This is one effect derived from the consideration of only one metric in the case of the SM ranking. In the case of MABRES, the consensus reached by experts (see Table 8.6) agrees that *Recall* is the most important metric (58% of the importance is attributed to it), but they do not neglect the contributions of the other metrics, which leads to the analyzed situation.

Something similar occurs in the *Best Effort* scenario. In this case, the SM method ranks T01 as the best alternative followed by T06, while MABRES ranks T06 as the best alternative followed by T01. Indeed, MABRES only sees a difference of 5% among first two alternatives, while their *F-measure*, the metric selected by the SM approach to rank alternatives in the *Best effort* scenario, has a more than 13% difference.

Going back to our analysis, we see that the higher differences in the proposed rankings appear in the *Heightened-Critical* and the *Minimum Effort* scenarios. In the first one, the SM ranking selects T06 as the best alternative, leaving T01 in the fifth position, while

MABRES ranks T01 in first position and T06 in the third. This can be explained looking into the values obtained in the metrics, where even though the fact that T06 outperforms the rest of tools when considering *Informedness*, it does not behave very well with *Recall*. In the case of considering a single measure this is not important since the 100% of the score depends on that metric. Conversely, in the case of MARBRE, the relative weight of metrics in each scenario becomes of major importance when computing scores. In the particular case of the *Heightened-Critical* scenario (see Table 8.6), *Recall* contributes with 32% of the score, while *Informedness* only provides 22%. This explains why a system with good numbers in both metrics outperforms another one despite the fact of not being the best for any of the considered metrics.

In the case of the *Minimum Effort* scenario, the variation in the ranking is not so critical, since it does not affect the winner alternative. The option ranked as second by the SM method (T09) is moved three positions down in the MABRES ranking, while T01, the seventh alternative when using the SM method, becomes the second one for MABRES. As in the previous case, the main problem is that, although T09 provides the best *Markedness* value, that values is rather low, so the alternative, which is not very good in *Recall*, *F-Measure* and *Precision*, is outperformed by others when considering the contributions to the score of all the metrics.

These results show the effect that neglecting part of the information provided by the available metrics has on the rankings, and also the contrary effect of considering the contribution of all metrics and the advice of experts.

8.5 Discussion

If we agree that several metrics are relevant for the proper characterization of benchmark targets, should we select an alternative based on the fact that it outperforms the rest in only one metric (certainly featuring only one of the aspects under consideration), or is it better to select the one providing the most balanced results? If a more balanced alternative is preferred, how can we properly determine such balance without the assistance of any expert or expert system?

Today, the benchmarking approaches for security tools based on binary classification (true/false positives/negatives) offer a simple way of comparing alternative solutions for different scenarios. One metric, from the whole set of possible metrics that can be derived from a confusion matrix (at least 14 were identified in [10]), is considered sufficient to rank the target tools. This approach is widely accepted, as it makes the decision process very straightforward and easy to understand. Nevertheless, simplifying the decision process by focusing on just one piece of information may outbalance its benefits. For example, having a look at the results presented in Table 8.7, tool T08 shows a perfect

Precision (a value of 1.00), which means that all the detected vulnerabilities are correctly classified. However, this same tool presents a very low *Recall* (a value of 0.019), which means that only a very small subset of the existing vulnerabilities are actually detected. If only *Precision* was to be considered as the metric to rank the tools, T08 would be ranked as the top alternative, but such ranking is obviously not good.

In dependability benchmarking, researchers usually require different metrics to obtain a holistic view of the system capabilities and, by analyzing them, work on the decision process. Obviously, this makes the analysis process more complex and cumbersome, in direct conflict with industry needs in terms of simplicity, but at the same time, the conclusions driven from the analysis are better-informed.

Our proposal offers an alternative approach to deal with this problem from the point of view of both academy and industry. On the one hand, the decision making process can be enriched by considering any number of metrics, thus leading to a better-informed decision. On the other hand, when our approach is applied, the decision is still made by using just a single score, thus keeping the simplicity in the decision making process. In fact, the underlying complexity of this approach is hidden from the benchmark consumer, as it resides on how this score is computed and not in how it is interpreted. But, *does this mean that the single metric approach is better as alternative tools are more widely separated and can be more easily discerned?* Probably not, and the most likely conclusion is that critical information missing in the single metric case is causing this effect. In spite of this, it is that internal complexity what can prevent the community from adopting the proposed approach, because trusting a complex non-familiar procedure instead of following the common and well-understood path is not easy.

Nevertheless, the *Single Metric* approach can be considered as a particular instance of MARBRES. This would be the situation where all the experts agree that only one metric is important to analyze the results in a given scenario, and the information provided by the other metrics is meaningless. The interesting fact is, that among the 84 pairwise comparison matrices obtained from the 21 experts (four pairwise comparison matrices per expert, one per scenario), not a single one of them showed a situation where one metric had a 100% importance, and the rest 0% (which is the case for the SM approach). From that point of view, MARBRES is able to encompass the type of analysis offered by SM, but it is not limited to it. It can also be used when a single metric is analyzed by multiple experts, when multiple metrics are considered by a single expert (maybe the benchmark user) and when multiple experts (not necessarily 21, but more than 2) analyze multiple (at least 3) metrics. The considered case studies have shown that i) MARBRES does not contradict in any scenario SM when a single metric is considered, and ii) it provides a fine grain analysis suitable to capture the important, and sometimes subtle, nuances existing between those alternatives with high scores. This is how the approach

enriches the information provided to benchmark users, thus letting them to take more informed decisions.

One can argue that there are many different approaches that can be used to increase the confidence that can be placed in the MABRES scores, especially regarding the weights obtained after processing the information provided by experts. For instance, the analysis of variance (ANOVA) may help in determining whether the mean of the responses provided by experts declaring a high expertise is statistically significantly different than that computed by experts declaring a low expertise. If this is the case, then it makes sense to restrict the data used to improve the confidence on the results provided at the cost of having less samples from which weights can be computed. However, in order to reach an acceptable level of statistical significance in the ANOVA analysis, the number of expert opinions to consider must be higher than ones considered in this paper. However, involving more than 21 experts in the analysis of benchmark results do not seem reasonable and it may be certainly prohibitive in terms of human (time) resources and money.

Likewise, it is also possible to compute the standard error associated to the geometric means used to obtain the consensus among experts. In such a way, this standard error will be propagated through the selected MCDM until reaching the final score. So, as a result, the score for each tool will be complemented by its standard error, defining an interval in which the actual score could be located. This will help the comparison of alternatives, since accordingly, tools with overlapping intervals could be considered as equally good, thus accounting for the bias/error experts could have introduced into their estimations. This is one of the lines of research that we are currently exploring for improving MABRES.

8.6 Conclusions

Benchmarking security tools is a process of prime importance not only to determine the most suitable tool to be used in a given application domain, but also to assess the effectiveness of any improvement deployed on existing tools. A large set of different metrics, focusing on a particular feature of these tools, have stemmed from the reported results in terms of true/false positives/negatives. These metrics are supposed to help benchmark users in better understanding the particular capabilities of each tool and, thus, reach a better decision. The truth is that, in practice, having to ponder and balance so many metrics, usually with conflicting goals, leads to multiple-objective optimization problems that are difficult to solve without any explicit guidelines. Accordingly, existing approaches usually opt to oversimplify the problem by considering just a single metric for each application scenario. Although this is an accepted practice by the industry, it is

also understood that the final decision may be biased by not considering all the subtleties accounted by the rest of neglected metrics.

This paper has proposed a novel and fully automated approach to alleviate the problem of simultaneously considering the contribution of all existing metrics towards the selection of the best security tool for a given application scenario. This approach relies on experts to judge the relative importance of each pair of metrics for each scenario. This process results in the quantification, according to each expert, of the contribution of each metric (weight) to the particular requirements of the selected scenario. However, as experts should make a decision as a group, their individual judgments are aggregated to reach an agreement that accounts for the individual contribution of each expert. These individual contributions can also be weighted according to the familiarity of the expert with the considered metrics and application scenario. Finally, the result of the proposed approach is a single score for each target tool that can be used to compare and select the best tool for the considered scenario.

Accordingly, this approach not only simplifies the decision process for the benchmark user by considering a single score, but also allows for a better informed decision as the contribution of all considered metrics towards the scenario requirements is taken into account in the process. Any bias that could be introduced by subjective judgments is minimized by considering the expertise of participants before reaching and agreement, and errors due to human intervention are also minimized by fully automating the whole process.

Future work relates to the use of expert systems and machine learning algorithms to complement in the medium term, and replace in the long term, the work currently assumed by human experts.

Chapter 9

Discussion of results

The results presented in previous chapters were focused on dealing with particular problems in the analysis of results in dependability benchmarking. This chapter summarizes and analyses such results according to their contribution to the methodology designed to improve the analysis of dependability benchmarking results.

9.1 Introduction

As stated in the first chapter of this thesis, the analysis process of dependability benchmark results has not been considered as part of the dependability benchmarking procedure, but instead as an external procedure that complements the former. Therefore, improving the process of analysis and comparison of results in this domain has been the primary goal of this work. Keeping in mind previous work carried out in this domain, the analysis process has been studied so it will adhere to the properties that a dependability benchmark should have, as they are defined in the DBench project [30].

The methodology defined in this thesis has decomposed the process of analysis into its elementary parts, and its interaction with the dependability benchmarking procedure has been redefined, so it could become part of it.

The following sections will discuss how this work has contributed to the definition of an analysis process that is compliant with those properties defined in the DBench project. Additionally, the definition of error-detection and error-correction mechanisms to verify the correct execution of the analysis will be addressed. Moreover, the versatility of this

methodology to adapt to the requirements imposed by different application domains is analyzed to determine how end-users can benefit from it. Nevertheless, this work does not affirm to provide *the perfect solution* for the problem of the analysis, but instead it provides an approach that tries to cover the problems identified up to date in the analysis of results in dependability benchmarks. Like any other approach, the methodology defined in this work has advantages and limitations, thus this chapter not only covers the advantages, but also the limitations, so the reader can make a better informed decision if she wishes to use it.

9.2 Satisfying dependability benchmark's properties

As stated in Chapter 1, dependability benchmarking procedures must be designed to satisfy a set of properties that can be verified and validated. This section discusses how the proposed methodology, which seamlessly integrates the analysis process into the common dependability benchmarking process, ensures that the required properties are met.

9.2.1 Non-intrusiveness

The notion of *intrusiveness*, in the context of the analysis, makes reference to any interference in the process that may have an impact on the conclusions.

In this work, as show in previous chapters, the intrusiveness in the analysis is dealt through the **definition** of the analysis process during the *specification* phase of the benchmark procedure. In order for the DM to define the analysis, it is necessary to know the type of output measures provided by the benchmark and its benchmarking context. Since this information is made available during the specification of the benchmark, the analysis can also be defined during this phase. By restricting its definition to this early phase in the process, the chances to interfere in the analysis are limited, and so **the analysis process is defined according to the requirements imposed by the benchmark, and not according to the results obtained from it**.

Nevertheless, this methodology has been defined to be used with different MCDM methods, and as it happens with the *Analytic Hierarchy Process* (AHP), sometimes additional work is required to satisfy this property. The AHP requires the DM to pairwise compare the results from the available alternatives for each criterion, and quantify the intensity of the importance between them. As a consequence of these comparisons of the results, a DM could (willingly or not) benefit one alternative over the rest by adjusting the values assigned to these intensities.

Since this pairwise comparison of values is necessary to apply the AHP, but the interaction of the DM may lead to a problem of intrusiveness, an **Assisted Pairwise Comparison Approach** (APCA) has been developed in this work. This approach, detailed in Chapter 6, automates the pairwise comparison process to avoid the interaction of the DM, thus preventing judgmental decisions from interfering with the analysis. Indeed, this approach not only contributes to satisfy the non-intrusiveness property for the analysis, but also guarantees that the analysis can be repeated for the same results, and therefore reproduced for other experiments.

Safeguarding the *non-intrusiveness* in the analysis is fundamental for third party users to trust the results obtained from dependability benchmarks. If the interaction of the DM is not limited only to the definition of the analysis, before the data is available, it could be thought that the analysis has been altered to benefit some conclusions.

9.2.2 Repeatability and reproducibility

One of the main goals of *dependability benchmarking* is to compare systems according to their behavior in presence of faults. In order to be able to compare the behavior of different systems, the dependability benchmarking procedure must be *repeatable* and *reproducible*. If these properties are not satisfied, then results cannot be trusted.

The same principle applies for the analysis process of dependability benchmark results. Its main goal is to analyze the results obtained from the benchmark experiments to characterize the behavior of the system through a single score, thus allowing the comparison of benchmarked systems.

Therefore, being able to guarantee that the process of analysis is *repeatable and reproducible* is key to use dependability benchmarks not only for assessment purposes, but also to be able to compare computer-based systems in presence of faults. If dependability benchmark results have to be compared between works, it is necessary for the analysis process followed to characterize the behavior of the system in a clear and explicit manner, so anyone can repeat or reproduce it. The methodology presented in this thesis guarantees the repeatability and reproducibility of the analysis process.

In the methodology defined in this work, different elements are used to define the analysis process in a formal and unambiguous way. First, the *aggregation of measures* defines how the final measures of the benchmark are hierarchically aggregated together until a single criterion that characterizes the behavior of the system is obtained. Then, the *requirements for the analysis* that are imposed by the benchmark context must be mapped into the analysis to provide it with the necessary context, thus leading to more meaningful conclusions. These requirements will determine which of the evaluated criteria are more relevant than others for that context, and by means of weights it can be quantified, and

therefore made explicit, the contribution of each criterion to the analysis. Additionally, the normalization procedures used to homogenize the metrics for their use in the analysis are also made explicit through the *scaling of metrics*. All these attributes conform what has been named in this work as the *quality model*. There is yet another aspect of the analysis process that is explicitly defined in this methodology, the *MCDM method* that will be used to perform the analysis.

Hence, the use of this methodology guarantees that the process of analysis can be repeated and reproduced by anyone, which will be necessary to expand the use of dependability benchmarks among people thus enriching cross-comparison of results among works.

9.2.3 Representativeness

In a dependability benchmark, the notion of *representativeness* is bound to how benchmark attributes reflect the actual conditions of a real scenario. In the context of the analysis of results for dependability benchmarks, the notion of representativeness acquires a distinct connotation. An analysis can be considered representative for a particular dependability benchmark and a given benchmark context, only if it is accepted by those using this benchmark to assess and compare a set of systems. This fact makes that the *representativeness* is possibly the hardest property to achieve for the analysis process, since it will always have a certain degree of subjectivity.

The heterogeneity among benchmark user's profiles implies that what one might consider a representative analysis, another may not. Commonly, people with a more academic (or research) profile may prefer to access all the raw data to perform a more in-depth analysis of the results. On the contrary, in an industrial context, where time is usually a highly valuable resource, using a single score to quickly compare systems might be preferable. To deal with this differences among user profiles, this work proposes the use of a hierarchical approach to characterize at different levels the behavior of benchmarked systems. This approach provides a complete analysis of the results that can be inspected from different perspectives, letting end-users to benefit from the approach that best suits their needs.

Nevertheless, the representativeness is conditioned by other features, like the capacity of the analysis process to grasp the requirements imposed by a particular benchmark context. Therefore, being able to represent such requirements, which are also in part determined by the objectives of the DM, is key to define a representative analysis, which will directly affect the representativeness of the dependability benchmark. As aforementioned, the use of weights in this work determine the contribution that each criterion has to characterize the behavior of the benchmarked system, thus representing in an explicit

manner the requirements of the benchmark context. Since the analysis is decomposed into smaller aggregations of criteria, which conform a hierarchical representation of the problem, criteria are weighted according to their contribution to the direct upper-level criterion, which eases the weighting process for the DM.

Nevertheless, the main difficulty relies on guaranteeing that the DM's objectives for the analysis are satisfied, as for the same context, opinions between DMs might differ regarding the contribution of each criterion. This problem becomes more obvious when the systems assessed by a dependability benchmark can be used in different contexts of application, as the data must be interpreted differently among contexts. It has been observed in this work that for a particular benchmark context, the quantification performed by different experts in the same field of the contribution of each criterion for the analysis, does not usually turn out to be exactly the same. Since experts apply their individual preferences and expertise to weight the criteria, slight (sometimes big) differences can be found in their decisions.

The methodology defined in this work applies techniques from the MCDM discipline designed to calculate, from the quantifications of several DMs, a unified and consensual set of requirements for the analysis. These techniques allow to quantify, in relative terms, the contribution of each expert to the consensual solution, being more representative to achieve a consensus the opinions from those with higher expertise than those with lower. Therefore, this approach is meant to define quality models for the analysis that can be considered *representative* by multiple experts. If the acceptance in the analysis process grows among experts, the chances for that analysis to be accepted by others and therefore considered *representative* will increase.

9.3 Error detection and correction in the analysis

While integrating the analysis process within the dependability benchmarking procedure, this work has guaranteed that the properties defined in the DBench project are not affected and also that they are satisfied in the analysis. The decomposition of the analysis in different parts that can be tackled individually, and the definition of procedures to deal with them, eases the definition of a clear and explicit analysis while guaranteeing that it is *non-intrusive*, *representative*, *repeatable* and *reproducible*. However, the satisfaction of these properties is related to the **definition** of the analysis, so it cannot be guaranteed that when the analysis is implemented, it will be free of errors.

The nature of dependability benchmarks is to assess both, functional and non-functional attributes of the systems, in presence of faults. Then, the amount of heterogeneous measures provided by the benchmark can be large, leading to lots of scaling, weighting and aggregation operations. This situation increases the complexity of the analysis, which

translates into possible sources of errors that can be made during the implementation of the analysis.

In order to detect those errors, the proposed approach, described in Chapter 5, is based in a well known technique used for testing the correct behavior of systems, *back-to-back* test. The proposal consists in verifying the correctness of the implementation done from the *definition* of the analysis process, and it is done by comparing its results with those from a secondary analysis. This alternative analysis is implemented from the same definition, which means that the same quality model is used, but this time, an alternative MCDM method is applied. Given the differences among MCDM methods, it is necessary that the method selected to implement the alternative analysis shares some similitudes with the main method.

From the application of this approach using the AHP and LSP it was possible to detect implementation errors that had an impact on the final classification of alternatives provided by the analysis. Since this back-to-back test approach has been specifically designed for this work, it takes benefit from the fact that the analysis is structured in a hierarchical way. The step-by-step procedure of the approach makes it possible to track an error down to its source, and in most of the cases it can be determined what type of error it is.

The higher complexity derived from the use of the proposed approach is mitigated by this *back-to-back* test approach, which enables DMs to verify that the implementation has been made according to the definition of the analysis, and no errors are present.

9.4 Application in multiple domains

The application of this methodology has been tested in several case studies from different domains in the field of dependability benchmarking of computer-based systems. Its flexibility to adapt to distinct type of metrics and schemes for the analysis has been demonstrated with these experiments, as well as how the analysis can be improved by providing a procedure able to meet the properties expected from a dependability benchmark.

In juxtaposition to commonly used methods, like the geometric mean, this methodology showed that it is able to map the evaluator's requirements for the analysis. Being able to consider the context of the benchmark in the analysis leads to providing end users with more meaningful conclusions. This fact is particularly important since benchmarked systems are assessed in presence of faults, and there will be situations where bad results in some criteria might imply severe consequences.

In addition, this methodology proved to be useful to define the analysis in a clear and explicit manner. Its application to case studies from other works, where the analysis was not clear or unambiguous, proved that those same analyses could be made clear and explicit, satisfying the *repeatability* and *reproducibility* properties in the analysis process.

Even more, the application of the *Assisted Pairwise Comparison Approach* (APCA) developed in this work proved to be useful to remove the human interaction from the application of the analysis. Designed to automate the process of pairwise comparison of results required by the AHP, its use in a case study evinced how it is possible to make the analysis *non-intrusive*, even when the MCDM method used requires otherwise. Actually, the use of MCDM techniques with this methodology also showed that it is possible to define analysis processes for dependability benchmarks that can be more likely to be accepted. With this methodology it is possible to define analysis that are the result of the consensual decisions of multiple experts in the same field of research, and so the *representativeness* of such analyses will be more difficult to be questioned.

Nevertheless, from its application in multiple case studies in different domains it could be seen that, despite the benefits mentioned to perform the analysis of results in dependability benchmarking, this methodology also presents some limitations. These limitations are discussed in the upcoming section to provide the reader with some insight on the issues that someone would have to face when applying this methodology.

9.5 Limitations

Despite the benefits that this methodology provides to deal with the analysis of results in dependability benchmarking, it also introduces more elements to the analysis, which in turn increases its complexity. The wide range of possibilities when defining all these elements and their intrinsic peculiarities, creates a set of limitations during its application.

9.5.1 Selection of the MCDM method

As already mentioned in this thesis, there is a large set of *MCDM* methods available to deal with multi-criteria decision problems [63]. In addition to this, deciding which of all these methods is the best one to deal with this kind of problems is in itself a multi-criteria decision problem, which creates what is known in the field of *operational research* as the *multi-criteria decision-making paradox* [108]. Hence, despite that one advantage of this methodology is that it is not bound to a particular MCDM method, which makes it more flexible, this could also be perceived as a disadvantage by end users, as deciding among the available methods can be overwhelming.

Along this thesis three different MCDM methods have been used to show the independence of the methodology from particular methods, although some guidelines were provided on how to select it. However, the decision of which MCDM method to use entirely depends on the DM, and its requirements about the analysis. For example, it has been mentioned that the AHP is a method that requires the values of all the alternatives to be available to perform the analysis. Hence, if another analysis is performed to other alternative, it would be necessary to have the data of all previous alternatives to compare their results against those obtained with the new analysis, since it performs relative comparisons. The WSM and the LSP on the other hand, both of them perform the analysis for each alternative individually, providing for each one a single score that is used to compare them. But there are other methods, like the *Weighted Power Model* (WPM) [109], that could be used with the proposed methodology to perform the analysis, that does not provide scores for the alternatives. As the AHP, this method performs relative comparisons among alternatives, but its mathematical process is based on ranking alternatives two-by-two from their results in all criteria. The alternatives are then sorted using these two-by-two rankings, which provides a global ranking of the alternatives, without scores involved.

Therefore, deciding which MCDM method to use can be a hard part when this methodology is applied. Actually, this may lead to a situation where each evaluator use the MCDM method that best suits her interests, or that she is more familiar with, avoiding to use the same method as others. But, despite the use of different MCDM methods among evaluators, the use of this methodology enables to share aspects like the quality model, that can be used by many, which is an improvement towards the cross-exploitation of results among works.

However, the use of the same quality model across works requires from the one thing that cannot be granted with this methodology, the acceptance among evaluators and DMs, which is covered next.

9.5.2 Acceptance of the process of analysis

The acceptance of a dependability benchmark among researchers and people from the industry is key to spread their use, which is necessary to examine and cross-compare results among different works.

This methodology is designed to integrate the process of analysis into the dependability benchmarking procedure, so it can assess and characterize the behavior of computer-based systems into a single score, easing the comparison among systems. Therefore, not only the dependability benchmark procedure needs to be accepted by others to open the

door to the cross-comparison of results among works, also the process defined to analyze such results.

In this methodology, techniques from the field of operational research have been applied to combine the expertise of different evaluators to reach consensus quality models for the analysis. Here, the core of the analysis, what makes it relevant for a given context, is decided by a set of experts in the same field of work, instead of being defined by only one evaluator. It seems reasonable to believe that this will contribute to expand its use, making it more likely to be accepted by others.

The bottom line here, is that despite defining several quality models to target different contexts for the analysis, and combining experts opinions to do so, it is not possible to guarantee that evaluators will accept the analysis, and this remains the problem. This is something that cannot be solved in spite of the methodology used to perform the analysis. The opinions among experts might differ regarding the analysis for the same context, and their individual acceptance of the consensus quality models driven from all their opinions cannot be assured. Nevertheless, the use of these procedures represents a great contribution towards achieving the acceptance of the analysis. Promoting the cooperation and discussion among experts to reach quality models that satisfy a majority, will make dependability benchmarks and their process of analysis more likely to be used by other people, which will ultimately impulse the sharing and cross-comparison of results among works.

Chapter 10

Conclusions and future work

10.1 Conclusions

The work done in the DBench project set the foundations for many works in the field of dependability benchmarking. The guidelines defined in this project contributed to the definition of many dependability benchmarks in different domains. However, it seems that while most of the research was done on improving the aspects related to the assessment of systems' behavior, little was done regarding how the measures that characterize those systems should be analyzed.

Unlike what happens with the dependability benchmark procedures, the process of analysis lacks of guidelines or standard procedures for its definition. This creates a situation where the analysis of dependability benchmark results is considered as something external to the benchmark itself, being left at the hands of end users. In those works where the analysis is actually performed, it either remains ambiguous, or fails to characterize the behavior of a system into a single score. It is common to see analysis carried out on the basis of each considered criterion, but rarely from a global perspective, where all the evaluated criteria are considered. In those cases where all criteria are considered, results are simply averaged, which limits in practice the capacity of the conclusions to reflect a real situation. These situations evidence that there is a need for methodologies able to contextualize the analysis to different scenarios, and that is capable of considering mul-

tiple heterogeneous measures to characterize the behavior of a system through a single score.

This thesis faces this problem to **improve the process of analysis and comparison of results in dependability benchmarking for computer-based systems**. Since dependability benchmarks have an intrinsic goal of providing means to compare alternative systems, the methodology proposed in this work has been developed around a main principle: *The analysis process must be considered as part of the dependability benchmarking procedure and so, it must comply with the same properties*. Therefore, the properties defined by the DBench project (*representativeness*, *non-intrusiveness*, *repeatability* and *reproducibility*) have been reinterpreted from the perspective of the analysis.

This methodology for the analysis has been designed so it can be integrated within the actual phases of a dependability benchmark procedure (see Figure2.1 in Chapter 2). The complete **definition** of the analysis during the *specification* phase of the benchmark assures the explicitness of the analysis process. For the analysis to be *repeatable* and *reproducible* it is necessary that all the elements that are part of the analysis process are made explicit.

This process is carried out through what has been labeled in this work as the **quality model** of the analysis. It represents the backbone of the analysis process, and its definition contributes to make explicit, and clear, the features of the analysis as defined by the DM: **i)** How the benchmark results are aggregated to calculate a single score that will characterize the system's behavior, **ii)** quantify the contribution of each criterion to calculate the aggregated scores, and **iii)** the individual procedure followed by each criterion to homogenize the results into the same units and scales.

Its early definition during the specification phase of a dependability benchmark is meant to preserve the objectivity in the analysis done by the DM. If the analysis was to be defined after the results from the experimentation were available, the DM could, willingly or not, define an analysis to meet preconceived expectations about a system's behavior. That kind of *intrusion* in the analysis by hands of the DM, even if not intentional, would lead to biased conclusions when assessing and comparing different systems. By restricting the definition of the analysis to the specification phase, it can be stated that the methodology presented in this work provides the means to assure a *non-intrusive* analysis procedure for a dependability benchmark. The aspects of that definition though, will determine the capacity of the analysis process to be *representative*, understanding the representativity of the analysis from two different perspectives: **i)** The acceptance of other users of the dependability benchmark to apply the defined analysis process, and **ii)** its capacity to reflect the requirements imposed by the application context of the benchmark.

The acceptance of an analysis process by other users will entirely depend on whether it is useful for them to carry out the assessment and comparison of different systems. By decomposing the problem through a *hierarchical* approach, the quality model is designed to deal with the different type of profiles present among benchmark users. In the industry, time is a valuable resource, thus users with an industrial profile might prefer to use a single score to compare benchmarked systems and reduce the time invested. While users with an academic profile, in addition to that single score, might prefer to access all the data from the analysis to study that data from different points of view. This hierarchical approach lets end users inspect the results of the analysis from different levels of abstraction. Analysis results can be observed either from a coarse grained point of view, where systems are compared through a global score, or from a fine grained point of view, where all intermediate results can be accessed.

In the other hand, for the results provided by the analysis to be meaningful for end users, it is necessary to use mechanisms that let DMs map the *requirements for the analysis* imposed by the context. With this methodology, DMs can define the relative contribution that each assessed criterion has to calculate the global score that will characterize a system's behavior. End users can therefore have a better understanding of the direct relation between experiment results and the final score. By understanding the process, its adequacy to a given context can be reasoned and discussed, and these are necessary elements to gain the acceptance of benchmark users.

The main problem with "acceptance" is that it cannot be taken for granted, and what a DM might consider a perfect analysis, it might not be seen as such by others. But in order to promote the cross-comparison of results among works, it is necessary that users agree to compare their results following the same analysis process. In this sense, this work proposes the use of techniques from the field of *operational research* to define, through consensus, the *requirements for the analysis*. Such techniques interpret and combine the individual requirements defined by a set of DMs (preferably experts in the field) into a single set of requirements. Since one of the objectives of this work is to promote the cross-comparison and sharing of results among dependability works, the use of these consensus requirements seem promising. Those analysis process defined through these techniques will present a better chance towards the acceptance by different users to carry out the same analysis than if they were defined by a single DM.

Meanwhile the **quality model** defines the structure of the analysis process, the mathematical procedures to actually compute the analysis of the results is carried out in this work through *multi-criteria decision-making* methods. Widely used in different domains, these methods are used in this work for their feasibility to be integrated within the methodology defined in this work. As there exists mathematical differences among the internal procedure in these methods, it is necessary to make clear which method will be used during the definition of the analysis. These differences among them can cause that some

methods might interfere with the properties that must be satisfied during the analysis. Nevertheless, that does not mean that a method is not suitable for this methodology, but that some extra work is required. This work has shown through the development an *Assisted Pairwise Comparison Approach* (APCA) how it is possible to cope with the problems that the AHP method presented to achieve a *non-intrusive* analysis.

Although, the internal procedures of the MCDM methods used are not the only possible source of conflict to provide accurate and error-free analysis. Compared to commonly used methods like the arithmetic or geometric mean, as the methodology defined in this work is able to provide more elaborated analysis, it also introduces many more features. An increase in the number of criteria that must be assessed to characterize the system might increase the size of the hierarchy, which consequently will increase the quantity of weights that need to be set. The higher the number of elements that need to be defined is, the higher the risk to make a mistake either during their definition, or during the implementation of the analysis.

Hence, this work proposes the use of a *back-to-back* test approach to verify the implementation of the analysis done from its definition. A second implementation of the analysis made with an alternative MCDM method let benchmark users inspect the hierarchical procedure of the analysis and detect possible errors made during the implementation. With this approach, when an error cause a difference in the global rankings, benchmark users are able to track it down to its source, and when possible, correct it.

From the work done, it can be stated that all the objectives defined for this thesis have been met, as well as its main goal, to **improve the process of analysis and comparison of results in dependability benchmarking for computer systems**. Despite that, there are different aspects that require from further research and are detailed in the next section, as they are the main ideas for future lines of work.

10.2 Future work

There are two main lines of work that would be interesting to pursue in order to complement the work done in this thesis. On one hand, a research needs to be carried out to assess the impact that the uncertainty present in some benchmarking measures has on the conclusions provided by the analysis. This kind of study can be useful for benchmark users comparing the results of different systems. It would be interesting to identify if variations on certain input values could modify the score obtained by a system, enough to produce a change in the conclusions provided by the analysis.

On the other hand, it has been mentioned in several occasions through this document that improving the analysis in this field of could have a significant impact in the cross-

exploitation and sharing of results among works. Therefore, to promote that situation, it would be convenient to develop a platform where this methodology is made accessible to benchmark performers, and where the cross-exploitation and sharing of results can be carried out.

10.2.1 Sensitivity analysis

The study done in [17] focuses on identifying the type of uncertainty present in the measures provided in dependability benchmarking. Despite the different types of uncertainty identified by the authors, there are some metrics that present a type of uncertainty that, as defined in that work, is considered as a *non-negligible* uncertainty. This type of uncertainty is related to those measures that quantify the dynamic behavior of the system, and therefore derive from continuous measurements performed in the system during a given period of time. In the *Guide to the Expression of Uncertainty in Measurement* (GUM [105]), this uncertainty is described as follows: **A parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to a measure**. This parameter, commonly expressed in the form of the *standard deviation*, represents a range of acceptable values that can be attributed to a measure. So, this type of measures present a certain degree of imprecision and changeability that can have an impact in the conclusions driven by the analysis.

Even though these type of measures are often present in the results of dependability benchmarks (like [49]), the truth is that their uncertainty is never considered in the analysis of results. Commonly used approaches in dependability benchmarking like the arithmetic mean, or the geometric mean, can only use a single value (and not a range of them) in their operations. The same thing happens when applying MCDM methods, as they are not meant to perform operations with values expressed in terms of their standard deviation. For that reason, sensitivity analysis for MCDM methods has increasingly widespread in many fields of engineering and sciences, becoming a necessary step to verify the feasibility and reliability of the conclusions driven by the analysis.

In the light of this problem, and knowing that a change in the input value of a measure can affect in some manner the final score of an alternative, a question comes up: **Can a value that changes within its range of uncertainty produce a change in the final score so the ranking of alternatives is affected?**. **Yes**, it can, the uncertainty present in the input values can have an impact on the conclusions, but since this is not true for every case, a sensitivity analysis is necessary to identify these situations.

Some evaluators may be tempted to think: *"I will not have problems as long as my benchmark does not have measures with non-negligible uncertainty"*. Well, that is not entirely true. Since the weights that quantify the contribution of the criteria in the analysis

are also input values of the analysis, variations in those weights can also have an impact in the conclusions. But, even though these weights do not have any implicit variation, the final score provided by the analysis not only depends on the input values of the measures, but also on the weights assigned to the criteria in the quality model. Therefore, variations in these weights (big or small) can alter the conclusions of the analysis.

Sensitivity analysis has been used in many research domains to study how the uncertainty in the input values of a mathematical model can affect the output. In the literature it can be found works like [58] where the sensitivity analysis is used to improve the performance of radioactive waste disposal, or [21], where it is used to enhance the economic assessment of health care technologies. However, the sensitivity analysis must be adapted to the mathematical procedure used by the analysis, so the scope of available methodologies is quite large [92]. In [111], the authors provide a very detailed description of the process that must be followed to perform a sensitivity analysis for the *Weighted Sum Model* (WSM) and the *Analytic Hierarchy Process* (AHP), both used in this thesis. The similarities between the operations performed during the aggregation process in both methods make it possible for the same kind of sensitivity analysis to be used in both situations.

Even though it is part of the ongoing work of this thesis, the first attempts to apply this type of analysis have proven to be beneficial to detect possible sources of uncertainty in the conclusions of the analysis. From a sensitivity analysis of the benchmark results, the minimum variation that must happen on a measure to produce a change in the conclusions can be quantified. Therefore, if measures with *non-negligible* uncertainty are present among the results, by comparing their uncertainty with the minimum variation required to change the conclusions, it is possible to identify which measure, in which alternative, presents a threat to the conclusions.

Assessing the sensitivity of the weights defined for the quality model of the analysis can provide interesting information about the relation between alternatives. By quantifying how much a weight needs to increase/decrease its value to modify the final ranking of alternatives, it can be determined if the quality model defined presents a threat to the conclusions obtained for a particular data set. It can happen that the minimum variation of a weight that would produce a change in the ranking of alternatives is very small. In that situation, the goodness of those alternatives changing positions in that ranking could be considered *similar*, and therefore the alternatives could be considered equivalent.

Up to date, the results show that the integration of sensitivity analyses would be an interesting contribution to the methodology defined in this thesis. Actually, in the field of dependability benchmarking, it seems that this would be the first time that such mechanisms are used to assess the impact that the uncertainty in the input values have in the conclusions. Even more, it could be a very important contribution towards the accep-

tance of dependability benchmarking procedures in many application domains by both, the industry and the research community .

10.2.2 An On-line accessible methodology

The methodology developed in this thesis presents benchmark performers with the means to define a process of analysis compliant with dependability benchmark procedures. Having a process of analysis that is unambiguous and explicit guarantees that it can be reproduced to perform a comparative assessment of dependability features across different systems. Although, the use of this methodology in dependability benchmarks cannot be an imposition, therefore, additional work is necessary in order to promote its use.

This line of work would be focused on implementing the methodology developed in this thesis in order to make it available for benchmark performers to use it. The main idea is to develop a framework where users can specify all the requirements of the analysis. By defining the measures provided by the benchmark, users will be able to build the hierarchical *aggregation of measures* for their analysis. This hierarchical aggregation would be used to map the *requirements for the analysis* by weighting the contribution of each criteria to the analysis. Then, from a set of already implemented normalization procedures, users would be able to determine which procedure should be used to perform the *scaling of metrics*, and configure it according to their needs.

The MCDM method to be used in the analysis would be selected from those already implemented in the framework. The option to introduce additional MCDM methods should be made available, either by request or by providing users with an API to interact with the system. An API would let users implement the methods, and upload them to the system after they have been reviewed, while a request would imply that the system administrator should do it.

Every process of analysis that is defined through the framework should be available for anyone to use them. By uploading their benchmark results, users would be able to specify not only the process of analysis, but to actually perform the analysis of their results. This would give them the chance to share their benchmarking results and the conclusions obtained from the analysis with the rest of users in the platform. Indeed, one of the objectives of this work is to let benchmark performers share the full specification of their dependability benchmark procedure. This would give others the chance to reproduce the same dependability benchmark experiments to assess other systems.

By using this platform, benchmark performers will be provided with an opportunity to impulse the cross-exploitation and sharing of results, and benchmarking procedures, in the field of dependability benchmarking. Having access to the full specification of dependability benchmarks defined by others would definitely give them more visibility in

the community. Benchmark performers could implement those benchmarks, assess different systems, analyze the results with the same analysis process and therefore compare their conclusions with those from other works.

The main problem that must be faced, is to convince researchers and people from the industry to get on board with the idea of sharing their work and results with others. However, this would represent a great chance for those users who participate to gain visibility for their work, and establish connections with other people working in the same field of research.

10.3 Related research activities

This section presents the several activities performed in relation to the research carried out during the development of this thesis, which include the related scientific publications, research stays, research projects and scientific speeches.

10.3.1 Related publications

The list of publications related to the work done in this thesis is divided into two categories, papers published in JCR journals and papers published in international conferences.

Publications in JCR journals

- Jesús Friginal, Miquel Martínez, David de Andrés and Juan Carlos Ruiz. "Multi-criteria analysis of measures in benchmarking: Dependability benchmarking as a case study". In: *Journal of Systems and Software*, vol. 111 (2016), pp. 105–118. (Impact Index 2016: 2.444, Q1)
  DOI: 10.1016/j.jss.2015.08.052

- Jesús Friginal, David de Andrés, Juan Carlos Ruiz and Miquel Martínez,. "A survey of evaluation platforms for ad hoc routing protocols: A resilience perspective". In: *Computer Networks*, vol. 75 (2014), pp. 395–413. (Impact Index 2014: 1.256, Q2)
  DOI: 10.1016/j.comnet.2014.09.010

- Miquel Martínez, David de Andrés, Juan Carlos Ruiz and Jesús Friginal. "From Measures to Conclusions Using Analytic Hierarchy Process in Dependability Benchmarking". In: *IEEE Transactions on Instrumentation and Measurement*, vol. 63.11 (2014), pp. 2548–2556.(Impact Index 2014: 1.79, Q2)
  DOI: 10.1109/TIM.2014.2348632

- Jesús Friginal, David de Andrés, Juan Carlos Ruiz and Miquel Martínez,. "REFRAHN: A Resilience Evaluation Framework for Ad Hoc Routing Protocols". In: *Computer Networks*, vol. 82 (2015), pp. 114–134. (Impact Index 2015: 1.446, Q2)
  DOI: 10.1016/j.comnet.2015.02.032

Publications in international conferences

- Miquel Martínez, David de Andrés and Juan Carlos Ruiz. "Gaining Confidence on Dependability Benchmarks' Conclusions through Back-to-Back Testing". In: *Tenth European Dependable Computing Conference (EDCC)*, Newcastle, 2014, pp. 130-137.
  DOI: 10.1109/EDCC.2014.20
- Miquel Martínez, David de Andrés, Juan Carlos Ruiz and Jesús Friginal. "Analysis of results in dependability benchmarking: Can we do better?". In: *IEEE International Workshop on Measurements & Networking (M&N)*, Naples, 2013, pp. 127-131.
  DOI: 10.1109/IWMN.2013.6663790
- Miquel Martínez, Yusheng Ji, David de Andrés and Juan Carlos Ruiz. "Assessment of Ad Hoc Routing Protocols for Network Deployments in Disaster Scenarios". In: *Workshop on Innovation on Information and Communication Technologies (ITACA-WIICT)*, Valencia, 2016, pp. 105–113.
- Miquel Martínez, David de Andrés and Juan Carlos Ruiz. "Comparing Benchmark Targets: Issues in the Analysis Model". In: *Workshop on Innovation on Information and Communication Technologies (ITACA-WIICT)*, Valencia, 2015, pp. 65–74.
- Miquel Martínez, David de Andrés and Juan Carlos Ruiz. "Multi-criteria decision-making techniques in dependability benchmarking: How to proceed?". In: *Workshop on Innovation on Information and Communication Technologies (ITACA-WIICT)*, Valencia, 2014, pp. 199–207.

10.3.2 Scientific research internships

During the development of this thesis, two research internships were done to promote the exchange of knowledge and cooperation between research groups:

- 06-01-2016 to 06-04-2016. *National Institute of Informatics, Tokyo, Japan*. Research supervised by Professor Yusheng Ji at the *Jusheng Ji Laboratory*. The research was focused on the dependability assessment, through simulation, and later analysis and comparison of results of ad hoc networks deployments in natural disaster scenarios.
- 01-09-2016 to 30-11-2016. *University of Coimbra, Coimbra, Portugal*. Research supervised by Professor Marco Vieira at the *Center for Informatics and Systems of the University of Coimbra*. The research was focused the analysis and comparison of results from the dependability assessment of vulnerability detection tools and web servers.

10.3.3 Research projects

The research carried out in this thesis was partially performed in the context of the following Spanish project:

- "**ARENES**: Adaptive and REsilient Networked Embedded Systems" under grant TIN2012-38308-C02-01. The main researcher in this project is Pedro Joaquín Gil Vicente, and the author of this thesis and its supervisors were part of the research team.

10.3.4 Scientific speeches

During the development of this thesis, I participated as co-speaker in a key note speech at the CyberCamp event of 2015, promoted by the *Spanish National Cybersecurity Institute* (INCIBE).

- Title: **Penetration testing in fourth generation SCADA systems**.
  Location: *CyberCamp* 2015, Madrid, Spain.
  Speakers: Jesús Friginal López and Miquel Martínez Raga.

References

[1] Yazeed A. Al-Sbou et al. “A Novel Quality of Service Assessment of Multimedia Traffic over Wireless Ad Hoc Networks”. In: *Proceedings of the 2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*. 2008, pp. 479–484. ISBN: 978-0-7695-3333-9 (cit. on pp. 42, 54).

[2] Raquel Almeida, Naaliel Mendes, and Henrique Madeira. “Sharing experimental and field data: the amber raw data repository experience”. In: *Distributed Computing Systems Workshops (ICDCSW), 2010 IEEE 30th International Conference on*. IEEE. 2010, pp. 313–320 (cit. on p. 50).

[3] José A. Alonso and M. Teresa Lamata. “Consistency in the Analytic Hierarchy Process: A new approach”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 14.04 (2006), pp. 445–459. DOI: 10.1142/S0218488506004114 (cit. on pp. 101, 137).

[4] Tiago L. Alves, José Pedro Correia, and Joost Visser. “Benchmark-Based Aggregation of Metrics to Ratings.” In: *IWSM/Mensura*. Ed. by Koichi Matsuda, Ken ichi Matsumoto, and Akito Monden. IEEE Computer Society, 2011, pp. 20–29. ISBN: 978-1-4577-1930-1 (cit. on p. 140).

[5] Ateret Anaby-Tavor, Avigdor Gal, and Alberto Trombetta. “Evaluating Matching Algorithms: the Monotonicity Principle”. In: *IIWeb*. 2003, pp. 47–52 (cit. on pp. 51, 52).

[6] David de Andres, Juan Carlos Ruiz, and Pedro Gil. “Using Dependability, Performance, Area and Energy Consumption Experimental Measures to Benchmark IP Cores”. In: *Forth Latin American Symposium on Dependable Computing (LADC)*. 2009, pp. 49–56 (cit. on pp. 42, 55).

[7] David de Andrés et al. “An Attack Injection Approach to Evaluate the Robustness of Ad Hoc Networks”. In: *IEEE 15th Pacific Rim International Symposium on Dependable Computing*. 2009, pp. 228–233 (cit. on p. 104).

[8] Leopoldo Angrisani and Michele Vadursi. “Cross-layer measurements for a comprehensive characterization of wireless networks in the presence of interference”. In: *Instrumentation and Measurement, IEEE Transactions on* 56.4 (2007), pp. 1148–1156 (cit. on p. 99).

[9] N. Antunes and M. Vieira. “Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach and Examples”. In: *IEEE Transactions on Services Computing* 8.2 (2015), pp. 269–283. ISSN: 1939-1374. DOI: 10.1109/TSC.2014.2310221 (cit. on pp. 127, 128).

[10] N. Antunes and M. Vieira. “On the Metrics for Benchmarking Vulnerability Detection Tools”. In: *The 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2015)*. Rio de Janeiro, Brazil: IEEE, 2015 (cit. on pp. 127, 129–131, 144, 149).

[11] Jean Arlat et al. “Fault injection for dependability validation: A methodology and some applications”. In: *IEEE Transactions on software engineering* 16.2 (1990), pp. 166–182 (cit. on p. 2).

[12] Luigi Atzori, Antonio Iera, and Giacomo Morabito. “The Internet of Things: A survey”. In: *Computer Networks* 54.15 (2010), pp. 2787 –2805. ISSN: 1389-1286 (cit. on p. 1).

[13] A. Avizienis et al. “Basic concepts and taxonomy of dependable and secure computing”. In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), pp. 11–33. ISSN: 1545-5971 (cit. on pp. 45, 98).

[14] Jonathan Barzilai and Boaz Golany. “Ahp Rank Reversal, Normalization And Aggregation Rules”. In: *INFOR: Information Systems and Operational Research* 32.2 (1994), pp. 57–64. DOI: 10.1080/03155986.1994.11732238.

eprint: http://dx.doi.org/10.1080/03155986.1994.11732238 (cit. on p. 137).

[15] Valerie Belton and Tony Gear. "On a short-coming of Saaty's method of analytic hierarchies". In: *Omega* 11.3 (1983), pp. 228–230 (cit. on p. 94).

[16] A. Bondavalli et al. "A New Approach and a Related Tool for Dependability Measurements on Distributed Systems". In: *IEEE Transactions onInstrumentation and Measurement* 59.2 (2010), pp. 820–831 (cit. on p. 40).

[17] A. Bondavalli et al. "Foundations of Measurement Theory Applied to the Evaluation of Dependability Attributes". In: *Dependable Systems and Networks, 2007. DSN '07. 37th Annual IEEE/IFIP International Conference on*. 2007, pp. 522 –533. DOI: 10.1109/DSN.2007.52 (cit. on pp. 3, 4, 58, 65, 167).

[18] Andrea Bondavalli et al. "A new approach and a related tool for dependability measurements on distributed systems". In: *Instrumentation and Measurement, IEEE Transactions on* 59.4 (2010), pp. 820–831 (cit. on p. 99).

[19] Sándor Bozóki, János Fülöp, and Attila Poesz. "On pairwise comparison matrices that can be made consistent by the modification of a few elements". In: *Central European Journal of Operations Research* 19.2 (2011), pp. 157–175. ISSN: 1435-246X. DOI: 10.1007/s10100-010-0136-9 (cit. on p. 114).

[20] J. L. Bredin et al. "Deploying Sensor Networks With Guaranteed Fault Tolerance". In: *IEEE/ACM Transactions on Networking* 18.1 (2010), pp. 216–228. ISSN: 1063-6692. DOI: 10.1109/TNET.2009.2024941 (cit. on p. 117).

[21] Andrew Briggs, Mark Sculpher, and Martin Buxton. "Uncertainty in the economic evaluation of health care technologies: The role of sensitivity analysis". In: *Health Economics* 3.2 (1994), pp. 95–104. ISSN: 1099-1050. DOI: 10.1002/hec.4730030206 (cit. on p. 168).

[22] P.S. Bullen. *Handbook of Means and Their Inequalities*. Mathematics and Its Applications. Springer, 2003. ISBN: 9781402015229 (cit. on p. 62).

[23] A. Ceccarelli. "Analysis of critical systems through rigorous, reproducible and comparable experimental assessment". PhD thesis. Università Degli Studi di Firenzi, 2012 (cit. on pp. 40, 50).

[24] Ming-Kuen Chen and Shih-Ching Wang. “The critical factors of success for information service industry in developing international market: Using analytic hierarchy process (AHP) approach”. In: *Expert Systems with Applications* 37.1 (2010), pp. 694–704 (cit. on pp. 13, 100, 134).

[25] Y. Chen, J. Yu, and S. Khan. “Spatial sensitivity analysis of multi-criteria weights in GIS-based land suitability evaluation”. In: *Environmental Modelling & Software* 25.12 (2010), pp. 1582 –1591. ISSN: 1364-8152. DOI: http://dx.doi.org/10.1016/j.envsoft.2010.06.001 (cit. on p. 65).

[26] Minsu Choi et al. “Reliability measurement of mass storage system for onboard instrumentation”. In: *Instrumentation and Measurement, IEEE Transactions on* 54.6 (2005), pp. 2297–2304 (cit. on p. 99).

[27] Giulio Concas et al. “Power-Laws in a Large Object-Oriented Software System”. In: *IEEE Transactions on Software Engineering* 33 (10 2007), pp. 687–708. ISSN: 0098-5589 (cit. on pp. 42, 54, 55).

[28] J. P Correia and J Visser. “Certification of technical quality of software products”. In: *Proceedings of the International Workshop on Foundations and Techniques for Open Source Software Certification*. 2008, pp. 35–51 (cit. on p. 55).

[29] Gordon Crawford and Cindy Williams. “A note on the analysis of subjective judgment matrices”. In: *Journal of Mathematical Psychology* 29.4 (1985), pp. 387 –405. ISSN: 0022-2496. DOI: http://dx.doi.org/10.1016/0022-2496(85)90002-1 (cit. on pp. 31, 135).

[30] *Dependability Benchmarking Project*. IST Programme, European Commission, IST 2000-25425, [Online]. Available: http://www.laas.fr/DBench. 2003 (cit. on pp. 2, 5, 6, 36, 40, 50, 53, 98, 153).

[31] Yucheng Dong et al. “A comparative study of the numerical scales and the prioritization methods in AHP”. In: *European Journal of Operational Research* 186.1 (2008), pp. 229 –242. ISSN: 0377-2217. DOI: http://dx.doi.org/10.1016/j.ejor.2007.01.044 (cit. on pp. 31, 135).

[32] Yucheng Dong et al. “Consensus models for AHP group decision making under row geometric mean prioritization method”. In: *Decision Support Systems* 49.3 (2010), pp. 281 –289. ISSN: 0167-9236. DOI: http://dx.doi.org/10.1016/j.dss.2010.03.003 (cit. on p. 137).

[33] A. Doupé, M. Cova, and G. Vigna. “Why Johnny Can’t Pentest: An Analysis of Black-Box Web Vulnerability Scanners”. In: *Detection of Intrusions and Malware, and Vulnerability Assessment*. Lecture Notes in Computer Science 6201. Springer Berlin Heidelberg, 2010, pp. 111–131. ISBN: 978-3-642-14214-7, 978-3-642-14215-4 (cit. on pp. 126, 127).

[34] Jozo Dujmović. “A method for evaluation and selection of complex hardware and software systems”. In: *CMG 96 Proceedings*. Citeseer. 1996 (cit. on p. 34).

[35] Jozo J. Dujmović and Henrik Legind Larsen. “Generalized conjunction/disjunction”. In: *International Journal of Approximate Reasoning* 46.3 (2007). Special Section: Aggregation Operators, pp. 423 –446. ISSN: 0888-613X. DOI: http://dx.doi.org/10.1016/j.ijar.2006.12.011 (cit. on p. 32).

[36] Jozo J. Dujmović and Hajime Nagashima. “LSP method and its use for evaluation of Java IDEs”. In: *International Journal of Approximate Reasoning* 41.1 (2006), pp. 3 –22. ISSN: 0888-613X. DOI: http://dx.doi.org/10.1016/j.ijar.2005.06.006 (cit. on pp. 27, 32, 44, 45).

[37] João Durães, Marco Vieira, and Henrique Madeira. “Dependability Benchmarking of Web-Servers”. In: *Computer Safety, Reliability, and Security*. Ed. by Maritta Heisel, Peter Liggesmeyer, and Stefan Wittmann. Vol. 3219. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pp. 297–310. ISBN: 978-3-540-30138-7. DOI: 10.1007/978-3-540-30138-7_25 (cit. on pp. 5, 40, 66, 67, 99).

[38] *EEMBC’s Benchmarks*. Embedded Microprocessor Benchmark Consortium, [Online]. Available: http://www.eembc.org/benchmark/products.php. 2014 (cit. on pp. 2, 4, 41, 53).

[39] Anikó Ekárt and S. Z. Németh. “Stability analysis of tree structured decision functions”. In: *European Journal of Operational Research* 160 (2005), pp. 676–695 (cit. on p. 85).

[40] I. A Elia, J. Fonseca, and M. Vieira. “Comparing SQL Injection Detection Tools Using Attack Injection: An Experimental Study”. In: *21st IEEE International Symposium on Software Reliability Engineering (ISSRE 2010)*. IEEE Computer Society, 2010, pp. 289–298. DOI: 10.1109/ISSRE.2010.32 (cit. on p. 127).

[41] İrfan Ertuğrul and Nilsen Karakaşoğlu. “Performance evaluation of Turkish cement firms with fuzzy analytic hierarchy process and TOPSIS methods”. In: *Expert Systems with Applications* 36.1 (2009), pp. 702–715 (cit. on pp. 13, 100, 134).

[42] European New Car Assessment Programme (EuroNCAP). *EuroNCAP*. [Online]. Available: http://www.euroncap.com/. 2013 (cit. on pp. 22, 52).

[43] *Evacuation Sites, Areas and Shelters in Central Tokyo*. http://www.realestate-tokyo.com/news/evacuation-sites-in-central-tokyo/ (cit. on p. 119).

[44] Tom Fawcett. “An Introduction to ROC Analysis”. In: *Pattern Recogn. Lett.* 27.8 (2006), pp. 861–874. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2005.10.010 (cit. on p. 128).

[45] Peter C. Fishburn. “Letter to the Editor - Additive Utilities with Incomplete Product Sets: Application to Priorities and Assignments”. In: *Operations Research* 15.3 (1967), pp. 537–542 (cit. on p. 29).

[46] Ernest Forman and Kirti Peniwati. “Aggregating individual judgments and priorities with the analytic hierarchy process”. In: *European journal of operational research* 108.1 (1998), pp. 165–169 (cit. on p. 108).

[47] Ernest Forman and Kirti Peniwati. “Aggregating individual judgments and priorities with the analytic hierarchy process”. In: *European Journal of Operational Research* 108.1 (1998), pp. 165 –169. ISSN: 0377-2217. DOI: http://dx.doi.org/10.1016/S0377-2217(97)00244-0 (cit. on p. 137).

[48] J. Friginal et al. “On Selecting Representative Faultloads to Guide the Evaluation of Ad Hoc Networks”. In: *Dependable Computing (LADC), 2011 5th Latin-American Symposium on*. 2011, pp. 94 –99. DOI: 10.1109/LADC.2011.18 (cit. on pp. 5, 35, 43, 44, 73).

[49] J. Friginal et al. “Using Dependability Benchmarks to Support ISO/IEC SQuaRE”. In: *IEEE 17th Pacific Rim International Symposium on Dependable Computing*. Pasadena, USA, 2011, pp. 28–37 (cit. on pp. 83, 167).

[50] Jesús Friginal et al. “A survey of evaluation platforms for ad hoc routing protocols: A resilience perspective”. In: *Computer Networks* 75 (2014), pp. 395–413 (cit. on p. 117).

[51] Jesús Friginal et al. “Coarse-grained resilience benchmarking using logic score of preferences: ad hoc networks as a case study”. In: *Proceedings of the 13th European Workshop on Dependable Computing*. EWDC ’11. Pisa, Italy: ACM, 2011, pp. 23–28 (cit. on pp. 81, 83).

[52] Jesús Friginal et al. “REFRAHN: A Resilience Evaluation Framework for Ad Hoc Routing Protocols”. In: *Computer Networks* 82 (2015), pp. 114–134 (cit. on p. 117).

[53] Romualdas Ginevičius. “Normalization of quantities of various dimensions”. In: *Journal of business economics and management* 9.1 (2008), pp. 79–86 (cit. on pp. 26, 27).

[54] Jim Gray. “A Measure of Transaction Processing 20 Years Later”. In: *CoRR* abs/cs/0701162 (2007) (cit. on p. 70).

[55] Jim Gray. *Benchmark Handbook: For Database and Transaction Processing Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992. ISBN: 1558601597 (cit. on p. 53).

[56] Jim Gray. *Benchmark Handbook: For Database and Transaction Processing Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992. ISBN: 1-55860-159-7 (cit. on pp. 126, 128).

[57] Salvatore Greco, J Figueira, and M Ehrgott. “Multiple criteria decision analysis: State of the Art Surveys”. In: *Springer's International series* (2005) (cit. on p. 12).

[58] Jon C Helton. “Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal”. In: *Reliability Engineering & System Safety* 42.2 (1993), pp. 327 –367. ISSN: 0951-8320. DOI: http://dx.doi.org/10.1016/0951-8320(93)90097-I (cit. on p. 168).

[59] Michael W. Herman and Waldemar W. Koczkodaj. “A Monte Carlo study of pairwise comparison”. In: *Information Processing Letters* 57.1 (1996), pp. 25

–29. ISSN: 0020-0190. DOI: http://dx.doi.org/10.1016/0020-0190(95)00185-9 (cit. on pp. 31, 135).

[60] *Hillsdale WMN*. Online: http://dashboard.open-mesh.com/overview2.php?id=Hillsdale. 2012 (cit. on pp. 82, 104).

[61] I. F. Akyildiz and X. Wang. "A survey on wireless mesh networks". In: *IEEE Communications Magazine* 43.9 (2005), S23–S30 (cit. on pp. 83, 104).

[62] International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). *ISO/IEC 25000. Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*. Geneve ISO. 2010 (cit. on pp. 22, 52, 56, 57).

[63] Alessio Ishizaka and Philippe Nemery. *Multi-criteria Decision Analysis: Methods and Software*. Wiley, 2013, p. 310 (cit. on pp. 117, 159).

[64] A. Jadhav and R. Sonar. "Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study". In: *2nd International Conference on Emerging Trends in Engineering and Technology*. 2009, pp. 991–997 (cit. on p. 29).

[65] Andrew Jaquith. *Security metrics: replacing fear, uncertainty, and doubt*. Upper Saddle River, NJ: Addison-Wesley, 2007. ISBN: 0-321-34998-9 978-0-321-34998-9 (cit. on p. 127).

[66] K. Kanoun et al. "Benchmarking the dependability of Windows and Linux using PostMark/spl trade/ workloads". In: *16th IEEE International Symposium on Software Reliability Engineering (ISSRE)*. 2005, pp. 10–20 (cit. on pp. 22, 52).

[67] Karama Kanoun and Lisa Spainhower. *Dependability Benchmarking for Computer Systems*. en. John Wiley & Sons, 2008. ISBN: 978-0-470-37083-4 (cit. on pp. 3, 9, 40, 41, 54, 80, 99, 126).

[68] Christopher W Karvetski, James H Lambert, and Igor Linkov. "Scenario and multiple criteria decision analysis for energy and environmental security of military and industrial installations". In: *Integrated Environmental Assessment and Management* 7.2 (2011), pp. 228–236. ISSN: 1551-3793. DOI: 10.1002/ieam.137 (cit. on p. 56).

[69] Murat Koksalan, Jyrki Wallenius, and Stanley Zionts. *Multiple Criteria Decision Making: From Early History to the 21st Century*. World Scientific Publishing Company; 1 edition (June 6, 2011), 2012. DOI: 10.1142/9789814335591 (cit. on pp. 81, 99).

[70] K. W. Kolence and P. J. Kiviat. "Software unit profiles and Kiviat figures". In: *ACM/Sigmetrics Performance Evaluation Review* 2.3 (1973), pp. 2–12 (cit. on p. 42).

[71] M. Król et al. "Extending Network Coverage by Using Static and Mobile Relays during Natural Disasters". In: *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. 2016, pp. 681–686. DOI: 10.1109/WAINA.2016.147 (cit. on pp. 116–118, 120, 123).

[72] Han-Lin Li and Li-Ching Ma. "Detecting and adjusting ordinal and cardinal inconsistencies through a graphical and optimal approach in AHP models". In: *Computers & Operations Research* 34.3 (2007), pp. 780 –798. ISSN: 0305-0548. DOI: http://dx.doi.org/10.1016/j.cor.2005.05.010 (cit. on p. 114).

[73] Nian Liu et al. "Security Assessment for Communication Networks of Power Control Systems Using Attack Graph and MCDM". In: *Power Delivery, IEEE Transactions on* 25.3 (2010), pp. 1492–1500. ISSN: 0885-8977. DOI: 10.1109/TPWRD.2009.2033930 (cit. on p. 56).

[74] Henrique Madeira, Marco Vieira, et al. "The OLAP and data warehousing approaches for analysis and sharing of results from dependability evaluation experiments". In: *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2003, pp. 86–86 (cit. on pp. 10, 54).

[75] J. Malczewski. *GIS and Multicriteria Decision Analysis*. Wiley, 1999. ISBN: 9780471329442 (cit. on p. 137).

[76] M. Martínez, D. de Andrés, and J.-C. Ruiz. "Gaining Confidence on Dependability Benchmarks' Conclusions through "Back-to-Back" Testing". In: *2014 Tenth European Dependable Computing Conference*. 2014, pp. 130–137 (cit. on pp. 56, 99, 121).

[77] M. Martínez et al. “Analysis of results in Dependability Benchmarking: Can we do better?” In: *M&N 2013, International Workshop on Measurements and Networking* (2013), pp. 127–131 (cit. on pp. 56, 81, 99, 121).

[78] M. Martínez et al. “From Measures to Conclusions Using Analytic Hierarchy Process in Dependability Benchmarking”. In: *IEEE Transactions on Instrumentation and Measurement* 63.11 (2014), pp. 2548–2556 (cit. on pp. 32, 56, 61, 122).

[79] Michele et al. “Power-Laws in a Large Object-Oriented Software System”. In: *IEEE Transactions on Software Engineering* 33 (2007), pp. 687–708. ISSN: 0098-5589. DOI: doi.ieeecomputersociety.org/10.1109/TSE.2007.1019 (cit. on p. 140).

[80] Michael F. Morris. “Kiviat graphs: conventions and figures of merit”. In: *ACM/Sigmetrics Performance Evaluation Review* 3.3 (1974), pp. 2–8 (cit. on pp. 42, 55).

[81] C. E. Perkins and E. M. Royer. “Ad-hoc on-demand distance vector routing”. In: *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA ’99. Second IEEE Workshop on*. 1999, pp. 90–100. DOI: 10.1109/MCSA.1999.749281 (cit. on pp. 118, 123).

[82] Charles E. Perkins and Pravin Bhagwat. “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”. In: *Proceedings of the Conference on Communications Architectures, Protocols and Applications*. SIGCOMM ’94. London, United Kingdom: ACM, 1994, pp. 234–244. ISBN: 0-89791-682-4. DOI: 10.1145/190314.190336 (cit. on pp. 118, 123).

[83] David Martin Powers. “Evaluation: from Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation”. In: *Journal of Machine Learning Technologies* (2011). ISSN: 2229-3981 (cit. on pp. 127, 129, 130).

[84] Amber project. *AMBER (Assessing, Measuring, and Benchmarking Resilience)*. FP7 Coordination Action, European Commission. 2008 (cit. on pp. 10, 54).

[85] Elizabeth M Royer and Chai-Keong Toh. “A review of current routing protocols for ad hoc mobile wireless networks”. In: *IEEE personal communications* 6.2 (1999), pp. 46–55 (cit. on p. 23).

[86] J.-C. Ruiz et al. “On Benchmarking the Dependability of Automotive Engine Control Applications”. In: *IEEE/IFIP International Conference on Dependable Systems and Networks*. 2004, pp. 857–866 (cit. on p. 40).

[87] J.-C. Ruiz et al. “On benchmarking the dependability of automotive engine control applications”. In: *Dependable Systems and Networks, 2004 International Conference on*. 2004, pp. 857–866. DOI: 10.1109/DSN.2004.1311956 (cit. on p. 99).

[88] Thomas L. Saaty. “Decision-making with the AHP: Why is the principal eigenvector necessary”. In: *European Journal of Operational Research* 145.1 (2003), pp. 85 –91. ISSN: 0377-2217. DOI: http://dx.doi.org/10.1016/S0377-2217(02)00227-8 (cit. on pp. 101, 114, 137).

[89] ThomasL. Saaty. “What is the Analytic Hierarchy Process?” In: *Mathematical Models for Decision Support*. Ed. by Gautam Mitra et al. Vol. 48. NATO ASI Series. Springer Berlin Heidelberg, 1988, pp. 109–121. DOI: 10.1007/978-3-642-83555-1_5 (cit. on pp. 43, 56, 61, 86, 88, 134).

[90] ThomasL. Saaty and LuisG. Vargas. “The Seven Pillars of the Analytic Hierarchy Process”. English. In: *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. Vol. 34. International Series in Operations Research & Management Science. Springer US, 2001, pp. 27–46. ISBN: 978-1-4613-5667-7. DOI: 10.1007/978-1-4615-1665-1_2 (cit. on pp. 31, 135, 137).

[91] T.L. Saaty. “Decision making with the analytic hierarchy process”. In: *International Journal of Services Sciences* 1.1 (2008), pp. 83–98 (cit. on pp. 30, 87, 100, 101, 121).

[92] Andrea Saltelli et al. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008. ISBN: 9780470725184. DOI: 10.1002/9780470725184 (cit. on pp. 65, 168).

[93] D. Saraswat and B.K. Chaurasia. “AHP Based Trust Model in VANETs”. In: *5th International Conference on Computational Intelligence and Communication Networks*. 2013, pp. 391–393 (cit. on p. 30).

[94] M. Sasaki et al. “Evaluation of communication system selection applying AHP algorithm in heterogeneous wireless networks”. In: *Computing, Communications and Applications Conference*. 2012, pp. 334–338 (cit. on p. 30).

[95] Gunnar Schröder, Maik Thiele, and Wolfgang Lehner. “Setting goals and choosing metrics for recommender system evaluations”. In: *UCERSTI2 Workshop at the 5th ACM Conference on Recommender Systems, Chicago, USA*. Vol. 23. 2011, p. 53 (cit. on p. 127).

[96] *SPEC's Benchmarks*. Standard Performance Evaluation Corporation, [Online]. Available: https://www.spec.org/benchmarks.html. 2014 (cit. on pp. 2, 66).

[97] B. Srdjevic et al. “Group Decision-Making in Selecting Nanotechnology Supplier”. In: *Nanomaterials: Risks and Benefits*. Ed. by Igor Linkov and Jeffery Steevens. NATO Science for Peace and Security Series C: Environmental Security. Springer Netherlands, 2009, pp. 409–422. ISBN: 978-1-4020-9490-3. DOI: 10.1007/978-1-4020-9491-0_32 (cit. on p. 108).

[98] S. V. Stehman. “Selecting and Interpreting Measures of Thematic Classification Accuracy”. In: *Remote Sensing of Environment* 62.1 (1997), pp. 77–89 (cit. on p. 128).

[99] D. Stuttard and M. Pinto. *The web application hacker's handbook: discovering and exploiting security flaws*. Wiley Publishing, Inc., 2007. ISBN: 978-0-470-17077-9 (cit. on p. 126).

[100] Stanley Y. W. Su et al. *A Cost-benefit Decision Model: Analysis, Comparison Amd Selection of Data Management*. Vol. 12. 3. New York, NY, USA: ACM, 1987, pp. 472–520. DOI: 10.1145/27629.33403 (cit. on pp. 32, 43, 62, 63).

[101] Xiaofeng Sun et al. “Construction and Operation of Analytic Hierarchy Process about Moral Education Evaluation in Colleges and Universities.” In: *Advances in Information Sciences & Service Sciences* 3.11 (2011) (cit. on pp. 13, 100, 134).

[102] T. Clausen and P. Jacquet. “Optimized Link State Routing Protocol(OLSR)”. In: *RFC 3626* (2003) (cit. on pp. 117, 123).

[103] *The network simulator NS-3*. https://www.nsnam.org (cit. on p. 118).

[104] *The SPEC Research IDS Benchmarking Working Group, officially called Benchmarking Architectures for Intrusion Detection in Virtualized Environments*. [Online]. Available: http://research.spec.org/en/working-groups/ids-benchmarking-working-group.html. 2013 (cit. on p. 80).

[105] SUBJECT TO CHANGE THEREFORE, MAYNOTBE AS, and A SAUDI STANDARD UNTIL APPROVED. “Guide to the Expression of Uncertainty in Measurement”. In: (1995) (cit. on p. 167).

[106] Louis L Thurstone. “A law of comparative judgment.” In: *Psychological review* 34.4 (1927), p. 273 (cit. on pp. 25, 101).

[107] *TPC’s Benchmarks*. Transaction Processing Performance Council, [Online]. Available: http://www.tcp.org/. 2013 (cit. on pp. 2, 4, 50, 53, 69).

[108] E. Triantaphyllou and S. H. Mann. “An examination of the effectiveness of multi-dimensional decision-making methods: a decision-making paradox”. In: *Decision Support Systems* 5.3 (Sept. 1989), pp. 303–312 (cit. on pp. 12, 94, 159).

[109] Evangelos Triantaphyllou. “Multi-Criteria Decision Making Methods”. In: *Multi-criteria Decision Making Methods: A Comparative Study*. Vol. 44. Applied Optimization. Springer US, 2000, pp. 5–21 (cit. on p. 160).

[110] Evangelos Triantaphyllou. *Multi-Criteria Decision Making Methods: A Comparative Study*. Vol. 44. Springer US, 2000. ISBN: 978-1-4757-3157-6. DOI: 10.1007/978-1-4757-3157-6 (cit. on p. 83).

[111] Evangelos Triantaphyllou and Alfonso Sánchez. “A Sensitivity Analysis Approach for Some Deterministic Multi-Criteria Decision-Making Methods*”. In: *Decision Sciences* 28.1 (1997), pp. 151–194 (cit. on pp. 65, 168).

[112] Marco Vieira and Henrique Madeira. “A dependability benchmark for OLTP application environments”. In: *Proceedings of the 29th international conference on Very large data bases - Volume 29*. VLDB ’03. Berlin, Germany: VLDB Endowment, 2003, pp. 742–753. ISBN: 0-12-722442-4 (cit. on pp. 40, 51, 69, 73, 99).

[113] *VIM3: International Vocabulary of Metrology - Basic and General Concepts and Associated Terms*. Joint Committee for Guides in Metrology, [Online]. Available: http://www.bipm.org/en/publications/guides/vim.html. 2012 (cit. on p. 99).

[114] M. A. Vouk. “Back-to-back testing”. In: *Information and Software Technology* 32.1 (1990), pp. 34–45. ISSN: 0950-5849 (cit. on pp. 34, 86).

[115] S. Wagner et al. “Comparing bug finding tools with reviews and tests”. In: *Testing of Communicating Systems* (2005), pp. 40–55 (cit. on pp. 126, 127).

[116] H Roland Weistroffer, Charles H Smith, and Subhash C Narula. “Multiple criteria decision support software”. In: *Multiple criteria decision analysis: state of the art surveys*. Springer, 2005, pp. 989–1009 (cit. on p. 11).

[117] W.T.M. Wolters and B. Mareschal. “Novel types of sensitivity analysis for additive MCDM methods”. In: *European Journal of Operational Research* 81.2 (1995), pp. 281–290 (cit. on p. 95).

[118] Minghui Wu et al. “COTS-based System’s Obsolescence Risk Evaluation”. In: *10th International Conference on Computer Supported Cooperative Work in Design*. 2006, pp. 1–5 (cit. on p. 29).

[119] R.R Yager. “A Note on Weighted Queries in Information Retrieval Systems”. In: *Journal of The American Society for Information Science* 38.1 (1987), pp. 23–24 (cit. on p. 62).

[120] AA Zaidan et al. “Multi-criteria analysis for OS-EMR software selection problem: a comparative study”. In: *Decision Support Systems* 78 (2015), pp. 15–27 (cit. on p. 12).

[121] Stelios H. Zanakis et al. “Multi-attribute decision making: A simulation comparison of select methods”. In: *European Journal of Operational Research* 107.3 (1998), pp. 507–529 (cit. on p. 95).

[122] A. Zanella et al. “Internet of Things for Smart Cities”. In: *IEEE Internet of Things Journal* 1.1 (2014), pp. 22–32. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2306328 (cit. on p. 1).

Acronyms

AHP	Analytic Hierarchy Process.
AIJ	Aggregation of Individual Judgments.
AIP	Aggregation of Individual Priorities.
AODV	Ad hoc On-demand Distance Vector.
APCA	Assisted Pairwise Comparison Approach.
BT	Benchmark Target.
CI	Consistency Index.
COTS	Commercial Off-the-shelf.
CR	Consistency Ratio.
DBench	Dependability Benchmarking project.
DM	Decision Maker.
DSDV	Destination-Sequenced Distance-Vector.
EEMBC	Embedded Microprocessor Benchmark Consortium.
EM	Evaluation Matrix.
FN	False Negatives.
FP	False Positives.
GCM	Group Comparison Matrix.
IDS	Intrusion Detection System.
IoT	Internet of Things.
LSP	Logic Score of Preferences.
MABRES	Multi-criteria Analysi of Benchmark Results with Expert Support.

MCDA	Multi-criteria decision analysis.
MCDM	Multi-criteria decision-making.
OLSR	Optimized Link State Routing.
OLTP	On-Line Transaction Processing.
OTS	Off-the-shelf.
QM	Quality Model.
REFRAHN	Resilience Evaluation FRamework for Ad Hoc Networks.
RGM	Row Geometric Mean.
SPEC	Standard Performance Evaluation Corporation.
SUB	System Under Benchmark.
TN	True Negatives.
TP	True Positives.
TPC	Transaction Processing Performance Council.
WMN	Wireless Mesh Network.
WSM	Weighted Sum Model.