

Doctoral (PhD) Dissertation

Dennis Weber

Kaposvár, Hungary

2026



HUNGARIAN UNIVERSITY OF
AGRICULTURE AND LIFE SCIENCES

A NOVEL APPROACH FOR PROPOSING AND ASSESSING COMPUTATIONAL TOOLS IN
MANUFACTURING INDUSTRIES

DOCTORAL (PhD) DISSERTATION

DENNIS WEBER

Kaposvár

2026

The PhD School

Name: Doctoral School of Economic and Regional Sciences
Hungarian University of Agriculture and Life Sciences,
Gödöllő, Hungary

Discipline: Management and Organizational Sciences

Head: Prof. Dr. Viktória Szente, PhD
Full professor
Institute of Agricultural and Food Economics
Hungarian University of Agriculture and Life Science

Supervisor: Dr. Mónika Varga, PhD
Senior researcher
Institute of Animal Sciences
Hungarian University of Agriculture and Life Science

.....
Approval of the Head of Doctoral
School

.....
Approval of the Supervisor

Content

LIST OF ABBREVIATIONS AND ACRONYMS	7
LIST OF FIGURES.....	8
LIST OF TABLES.....	10
1. INTRODUCTION.....	11
1.1 Background and motivation	11
1.2 Methodological overview	12
1.3 Significance of the research.....	12
1.4 Structure of the dissertation.....	13
1.5 Problems to solve	14
2. OBJECTIVES TO ACHIEVE	15
3. LITERATURE REVIEW	16
3.1 How applied artificial intelligence creates value for manufacturing industries	16
3.1.1 Engineering and management of developing sustainable manufacturing.....	16
3.1.2 Evolving methods and tools of artificial intelligence	17
3.1.3 AI/ML-driven value creation.....	18
3.1.4 AI applications for sustainable manufacturing – the connection between AI and sustainability	19
3.2 Systematic literature review to establish a knowledge base.....	21
3.2.1 Search term and database selection.....	21
3.2.2 Inclusion/exclusion criteria.....	22
3.2.3 Screening	23
3.2.4 Topical cluster identification	25
3.3 Literature in the context of how we can learn from the experts’ opinions	29
3.4 Methods for discovery of methodological combinations	30
4. MATERIALS AND METHODS.....	32
4.1 Computational problem-solving methods involved in the further classification.....	32
4.1.1 Conventional computational problem-solving methods explained.....	32
4.1.2 AI/ML-based computational problem-solving methods explained	36
4.2 The method of Reevaluable Possibility Space: an equivalence classification and evaluation feedback-based learning	39
4.2.1 Minimal and maximal combinations in a property lattice	40
4.2.2 Evaluation of the property lattice.....	41
4.2.3 Reevaluable Possibility Space for generation of problem-solving methods.....	43
5. RESULTS	47
5.1 Simplified illustrative example for possibilistic learning from the evaluated lattice of equivalence-classified properties	47

5.2	Equivalence-classified properties of the investigated set of computational problem-solving methods for sustainable manufacturing	58
5.2.1	Property classes and properties of manufacturing tasks.....	58
5.2.2	Property classes and properties of computational problem-solving methods	62
5.2.3	Interpretation of the property classes in the context of manufacturing systems	64
5.3	Initial compatibility structure of the binary relations between the property classes.....	65
5.4	Description of the complete combinations	67
5.5	Description of the binary relations between the classified properties	68
5.6	Overview of the algorithm in the sense of Reevaluable Possibility Space.....	70
5.7	Computational representation of the algorithm	71
5.8	Workflow of case studies.....	73
5.8.1	Construction of an initial empty knowledge base and the initial set of evaluated complete combinations	73
5.8.2	First reevaluation of minimal combinations.....	75
5.8.3	Aggregation of experts' knowledge	75
5.8.4	Proposition of new complete combinations	76
5.8.5	Expert evaluation of suggested new combinations	76
5.9	Evolution of the binary knowledge base.....	76
6.	DISCUSSION.....	80
6.1	Evaluation and interpretation of the suggested complete combinations.....	80
6.2	Evaluation of the final state of the binary knowledge base	86
6.3	Interpretation of the learning behavior of the RPS	90
6.4	Comparison with existing method discovery approaches.....	93
6.5	Reconnection of results to the literature	95
6.6	Practical implications.....	96
7.	CONCLUSIONS AND RECOMMENDATIONS.....	99
8.	NEW SCIENTIFIC RESULTS.....	102
9.	SUMMARY	105
10.	ACKNOWLEDGEMENTS.....	108
11.	DECLARATION	109
12.	APPENDICES	110
A1:	Bibliography	110
A2:	SWI-Prolog code of the simplified implementation of the RPS algorithm	119
A3:	Details about the simplified illustrative example for learning from the RPS	125
A4:	Related publications.....	139

LIST OF ABBREVIATIONS AND ACRONYMS

Table 1 List of abbreviations and acronyms

#	Abbreviation/ Acronym	Meaning
1	AI	Artificial Intelligence
2	ML	Machine Learning
3	TBL	Triple Bottom Line
4	OSHA	Occupational Safety and Health Organization
5	IT	Information Technology
6	IoT	Internet of Things
7	ANN	Artificial Neural Networks
8	CNN	Convolutional Neural Networks
9	SVM	Support Vector Machine
10	k-NN	k-Nearest Neighbor
11	GA	Genetic Algorithms
12	MPC	Model-based Predictor-Corrector
13	SPC	Statistical Process Control
14	MILP	Mixed Integer Linear Programming
15	MINLP	Mixed Integer Non-Linear Programming
16	FEA	Finite Element Analysis
17	CAD	Computer-Aided Design
18	CAM	Computer-Aided Manufacturing
19	SBO	Simulation-Based Optimization
20	ELS	Evolutionary Learning Strategies
21	RPS	Reevaluable Possibility Space

LIST OF FIGURES

Table 2 List of Figures

#	Figure	Page
1	Number of publications by year	24
2	Appearance of AI and ML in reviewed literature	24
3	Topical clusters by count of search term appearance in reviewed literature	25
4	Illustration of the changing evaluation of a given binary relation	50
5	Illustration of ranking of binary relations within a given class, according to the strategy maxmin	51
6	Summarized uncertainties of unknown and known binary relations, as well as the total uncertainty for evaluator e1, in case of 3 evaluators	52
7	Comparison of inherent uncertainty with the uncertainty of knowledge calculated in previous case study, made for simulated expert e1, in case of 3 experts with no consent	53
8	Uncertainty of available knowledge for 3 evaluators	54
9	Uncertainty of available knowledge for 8 evaluators	55
10	Various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators, utilizing stepwise consent	56
11	Effect of stepwise consensus, in case of 3 evaluators	56
12	Effect of stepwise consensus for various types of uncertainties, in case of 8 evaluators	57
13	Property classes and properties of manufacturing tasks	58
14	Property classes and properties of computational problem-solving methods	62
15	Relations among properties of P1 - P6 equivalence classes	66
16	Exemplary graphic representation of the binary combinations of properties of the property classes P1 and P4	67
17	Simplified scheme of the evolving knowledge base of evaluated binary relations	70
18	Development of the uncertainty of binary relations	79
19	Various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators	128
20	Uncertainty of lack of knowledge from 3 evaluators	128

21	Uncertainty of available knowledge for 3 evaluators	129
22	Total uncertainty for 3 evaluators	129
23	Various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators, utilizing stepwise consent	130
24	Uncertainty of lack of knowledge, in case of 3 evaluators, utilizing stepwise consent	131
25	Uncertainty of available knowledge, in case of 3 evaluators, utilizing stepwise consent	131
26	Sum of uncertainty, in case of 3 evaluators, utilizing stepwise consent	132
27	Various kinds of uncertainties of knowledge for evaluator e1, in case of 8 evaluators	133
28	Uncertainty of lack of knowledge from 8 evaluators	134
29	Uncertainty of available knowledge for 8 evaluators	134
30	Total uncertainty for 8 evaluators	135
31	Various kinds of uncertainties of knowledge for evaluator e1, in case of 8 evaluators, utilizing stepwise consent	136
32	Uncertainty of lack of knowledge, in case of 8 evaluators, utilizing stepwise consent	137
33	Uncertainty of available knowledge, in case of 8 evaluators, utilizing stepwise consent	137
34	Total uncertainty for 8 evaluators, utilizing stepwise consent	138

LIST OF TABLES

Table 3 List of Tables

#	Table	Page
1	List of abbreviations and acronyms	7
2	List of figures	8
3	List of tables	10
4	Initial search results in high numbers of findings	21
5	Results by database	22
6	Inclusion/exclusion criteria	22
7	Quantitative report on the evolution of the binary knowledge base	77
8	The ten highest and ten lowest evaluated newly suggested combinations of Rmax_DW1TH1SR1full1_650	80
9	Correlation between average Frequency and average Uncertainty in paired classes	86
10	Analyzing P1 Task of application in the latest knowledge base	88
11	Implementation Roadmap for the adoption of complete combinations	97

1. INTRODUCTION

1.1 Background and motivation

Manufacturing systems are undergoing a paradigm shift characterized by increasing complexity, dynamic market demands, and an imperative for sustainable and circular production. As manufacturers seek to navigate this evolving landscape, the adoption of Artificial Intelligence (AI), particularly Machine Learning (ML), has emerged as a critical enabler for process optimization, quality assurance, supply chain resilience, and resource efficiency. The integration of AI/ML into engineering and managerial decision-making in manufacturing holds the promise of augmenting traditional problem-solving methodologies with adaptive, data-driven, and scalable solutions.

Despite the growing body of literature on AI/ML applications in manufacturing, a critical gap persists. Amidst a multitude of new Artificial Intelligence (AI) and specifically Machine Learning (ML) methods and tools, as well as their combinations with traditional computational tools (Naser, 2025; Plathottam et al., 2023), identifying and matching workable solutions to various specific manufacturing tasks is challenging. According to our knowledge, there is no systematic overview for structuring different types of conventional and AI/ML applications across various levels and tasks of problem-solving. Most existing reviews and case studies focus on individual applications, technologies, or isolated use cases. What is often missing is a systemic classification that captures the multi-dimensional nature of AI/ML applications across the full spectrum of tasks, problem-solving phases, organizational levels, and performance objectives within manufacturing systems. That complex interrelation between conventional and AI/ML-based methods remains underexplored, hindering the development of holistic strategies for AI adoption.

Compared to traditional manufacturing objectives, sustainable manufacturing is subject to a different and extended set of challenges. Among them are trade-offs between economic, environmental, and social objectives; and effects extending beyond current operations, affecting long time frames, industries, and systems. Additionally, sustainable manufacturing is subject to the dynamic development of regulatory requirements, such as environmental standards, reporting obligations, and compliance frameworks.

This dissertation addresses this gap by proposing and demonstrating a knowledge-based, feedback-driven methodology to map and analyze the possibility space of AI/ML-based and conventional computational problem-solving methods applied to manufacturing tasks.

1.2 Methodological overview

The research begins with a comprehensive literature review to extract a wide range of conventional and AI/ML-based computational problem-solving methods in manufacturing. Through equivalence classification, six equivalence classes of properties are identified. These classes capture critical dimensions such as problem-solving phases, task types, performance objectives, levels of application, and methodological approaches. Between the properties of these classes, 15 compatibility relations and 733 binary relations are formally defined.

Valid solution strategies are described as complete combinations, each comprising one property from each class. Starting from a heuristically selected subset of complete (maximal) combinations evaluated by a group of experts, a feedback-based learning algorithm is implemented. This algorithm normalizes the real-valued assessments and maps them to [Min, Max] subintervals within the unit interval $[0, 1]$, associated with the binary (minimal) combinations. These subintervals represent uncertain knowledge about the value of minimal, binary combinations.

The algorithm then proposes new complete combinations that are tendentiously better performing, guided by the structured uncertainty representation. The process iterates between reevaluation and generation, allowing the knowledge base to evolve. Over time, the uncertainty of the initial ‘lack of knowledge’ decreases while the accumulated knowledge-based uncertainty increases, exhibiting a minimum in total uncertainty at an intermediate stage.

Expert opinions are aggregated using consistent rules, and their effects on the valuation and structure of the knowledge base are analyzed. The evaluation feedback mechanism ensures that evolving expert understanding continually shapes the possibility space and its interpretation.

1.3 Significance of the research

The methodology developed and tested in this dissertation contributes to both academic research and industrial practice. From a scientific perspective, it offers a novel approach to knowledge representation and learning in multi-dimensional decision spaces, grounded in expert evaluation and modelling of possibilistic uncertain knowledge behind this method discovery problem. The proposed framework provides insights into how AI/ML-based computational problem-solving methods can be systematically classified and combined with traditional approaches, addressing the growing complexity of manufacturing systems under sustainability constraints.

From a practical perspective, this work offers a decision-support tool for manufacturers and AI solution providers. By mapping the landscape of tasks and methods, the newly developed classification allows them to describe and better understand the interconnectedness of conventional and AI applications in manufacturing. This understanding aims to serve as guidance for manufacturers who want to evolve from early-stage, conventional, and isolated AI applications to systematic, integrated AI leverage.

Example industrial energies supply:

In an industrial park setting, electrical power, steam, compressed air, and cooling are supplied from a single infrastructure provider, each produced in a separately operated process. Multiple and different computational tools are in use in those production facilities. Demand for each industrial energy fluctuates; at the same time, the production facilities have limited volume control. Which computational problem-solving method or combination of methods can impactfully analyze integrated demand patterns, leveraging the existing systems, integrate the findings, and derive optimization recommendations while respecting the limitations of each production?

This analysis allows AI solution providers to gain expertise at the intersection of AI, conventional problem-solving methodologies, and manufacturing in the context of sustainability-driven complexity issues. The ability to iteratively learn and adapt from expert input reflects the dynamic nature of manufacturing environments and helps stakeholders align AI capabilities with real-world challenges.

1.4 Structure of the dissertation

The remainder of this dissertation is structured as follows: Chapter 2 details the objectives to achieve in this dissertation. Chapter 3 presents a detailed literature review of AI/ML applications in manufacturing and how learning from experts is operationalized. Chapter 4 introduces the methodology used to achieve the dissertation's objectives. Chapter 5 provides a comprehensive analysis of the results, including knowledge base evolution, uncertainty analysis and evaluation. A comprehensive discussion of the results is found in Chapter 6. Chapter 7 considers implications, limitations, and future directions. Chapters 8 and 9 conclude the dissertation, listing new scientific results and summarizing their potential impact on the manufacturing sector.

1.5 Problems to solve

The literature provides exemplary assessments of practical computational problem-solving methods and tools, based on real-life use cases. However, a manufacturing-focused methodology for learning from solutions in the scientific literature and suggesting possible new combinations of methods from an accumulated knowledge base does not exist.

This work aims to introduce and test a methodology for the systemic categorization and evaluation of conventional and AI/ML applications in manufacturing, in a way that supports a clear overview of the existing methodological combinations, as well as contributes to the automatic generation-based suggestion of new solutions from a continuously evolving knowledge base, derived from the existing methods.

2. OBJECTIVES TO ACHIEVE

The complex, circular systems of sustainable manufacturing require increasingly sophisticated methodologies, along with a clear systemic analysis of the underlying complex tasks and of appropriate combinations of conventional and AI/ML-based computational applications for the solution of these tasks. Literature based on real-life use cases provides exemplary assessments of usable methods and tools. However, what is needed is a methodology to learn from these solutions, as well as to suggest possible new combinations from the accumulated knowledge base.

This dissertation aims to provide an overview of various engineering tasks and tools, focusing on AI/ML methods and on the most current AI applications in manufacturing. A systemic perspective is added to the overview by structuring AI/ML applications in manufacturing along multiple dimensions, represented by equivalence classes of features, determining the tasks to be solved, the levels of application, the phases of problem solving, the objectives, as well as the various types of conventional and AI/ML-based methodologies. The resulting classification allows us to describe and better understand the interconnectedness of conventional and AI applications in manufacturing. Accordingly, the *main objective* of this work is to introduce and to test a methodology for the systemic categorization and evaluation of conventional and AI/ML applications in manufacturing, in a way that supports a clear overview of the existing methodological combinations, as well as contributes to the automatic generation-based discovery of new solutions from a continuously evolving, but uncertain knowledge base of the existing methods. The *specific aims* are the following:

- to classify the properties, characterizing
 - a) the engineering and management tasks, as well as
 - b) the applied conventional and AI/ML-based methods and tools used for developing sustainable manufacturing;
- to assess the usefulness of computer-supported problem solving in manufacturing, based on published case studies and experts' knowledge;
- to create a learning knowledge base and a methodology for the discovery of usable, advantageously AI-involved, optionally combined computational methods and tools for supporting the development of new, effective, and sustainable solutions.

3. LITERATURE REVIEW

3.1 How applied artificial intelligence creates value for manufacturing industries

3.1.1 Engineering and management of developing sustainable manufacturing

Definitions of manufacturing commonly include a description of required assets, the processing of materials, and resulting goods or products (Kakaomerlioglu & Carlsson, 1999; Segreto & Teti, 2014; Levinson, 2017).

Industrial manufacturing can be characterized by utility-scale production assets, allowing large volumes of product while consuming high amounts of energy, physical resources, and financial expenditures. Compared to smaller-scale manufacturing, high degrees of automation are implemented, and fewer human resources are required per unit of product.

Important to note, “manufacturing” is not a synonym for “production.” While production includes value-adding processes, manufacturing additionally encompasses organizational and managerial functions like design, process planning, and control.

Pursuing ever-greater competitive edge, process control, and optimization is at the core of industrial manufacturing. Process control and optimization balance output, quality, and consumed resources, including time and labor, while maintaining safety and compliance. Process optimization aims to eliminate unnecessary steps, schedules maintenance, and automates as much as possible (Michigan State University - Global EDGE, 2000).

Continually optimized manufacturing processes are progressing sustainability objectives, as they aim to reduce raw material use and energy consumption and increase efficiencies. Still, classic process optimization does not address all aspects of sustainability in the environmental, social, and governance space. Embedded in this larger context, manufacturing is impacted by supply chain disruptions, IT and data security issues, or dynamic regulatory conditions. Consequently, successful engineering and management of developing sustainable manufacturing must include a broader range of topics than classic process optimization to develop a sustainable, ideally circular economy against exhausting raw materials and accumulating waste. Systems are affected by demographic and climate change, as well as by fast shocks caused by extreme weather and unexpected ecological and socio-economic situations. Planning and operation of these large-scale complex processes require appropriate integration of diversified elements of field- and task-specific digitized data and models (Geissdoerfer et al., 2017; Kirchherr et al., 2017).

3.1.2 Evolving methods and tools of artificial intelligence

Recently, AI and especially ML “came into vogue,” as a methodological support for the analysis of Big Data generated by the Internet of Things (IoT) (e.g., intelligent data analytics for time series and images). It is worth remembering that AI had started with two challenges (e.g., Luger & Stubblefield, 1989; Chowdhary, 2020). The foundations of knowledge representation and search prepared expert systems and machine learning, on the one hand. The (practically first-order) predicate calculus initiated the development of AI programming languages (LISP and Prolog), on the other hand.

The idea of neural system inspired Artificial Neural Networks (ANN) was suggested (e.g., by McCulloch & Pitts, 1943) well before the foundations of AI. The further developed ANN, the biosystems-motivated evolutionary and genetic algorithms (Holland, 1975), as well as the cognitive science-inspired fuzzy set theory and logic (Zadeh, 1965) were grouped into Computational Intelligence, later resulting in combined applications of these three paradigms.

Engineering applications of AI were implemented throughout manufacturing industries, complementing the traditional computational methods of model-based design and control. However, the more complex systems fell behind the computer model-based problem-solving at that time. Recently, these applications met the opportunities of affordable sensors, the Internet of Things, Big Data, intelligent data analytics, etc., and progressed rapidly. Simultaneously, this fast development caused the forgetting of the nonlinear dynamic balance-based methods of engineering design, planning, control, and operation, as well as the loss of a clear overview of causal relations and balances behind complex process systems. Nowadays, inspired also by the sustainability of natural systems (e.g. Varga et al., 2023), the planning and operation of complex circular processes emphasizes the combined applications of AI with other engineering methodologies.

AI and ML are often used synonymously by people outside the technology industry. While there is some correlation between the two, they are quite different. AI generally refers to a program or system capable of performing tasks that typically require human intelligence. Some examples include facial recognition, image classification, weather forecasting, and language translation (Russel & Norvig, 2020).

ML, on the other hand, is a subset of AI that focuses on algorithms that can learn from their mistakes and improve their performance as they are used over time. In other words, if you give an ML program enough information about a certain feature of an object or subject, it will be able to identify the object or subject more accurately than when it started (Alpaydin, 2020). ML

algorithms are most used in manufacturing because they help machines make decisions based on data rather than programming them with every possible scenario they may encounter (Li et al., 2022).

A typical set of AI/ML-based computational problem-solving methods in manufacturing, considered in this work, includes the following methods and tools: Convolutional Neural Networks (CNNs) (Goodfellow et al., 2016); Support Vector Machines (SVMs) (Cortes & Vapnik, 2020); Random Forests (Breiman, 2001); k-Nearest Neighbors (k-NN) (Alpaydin, 2020); Fuzzy Logic and Reasoning (Ross, 2009); Genetic Algorithms (GAs) (Goldberg, 2013); Evolutionary Learning Strategies (Beyer & Sendhoff, 2008).

3.1.3 AI/ML-driven value creation

AI technology offers solutions not only for process optimization in manufacturing industries. Plastino and Purdy (Plastino & Purdy, 2018) identify three clusters of AI applications to increase profits independent of specific industries. The first cluster is intelligent automation, upscaling the idea of local monitoring to a wider dimension encompassing the whole supply chain. Instead of limiting the sensor-based monitoring to individual machines or production sites, intelligent automation leverages AI's advantages from procurement to customer service. Spot-market priced sourcing, interdependent productions, competitor and market intelligence, production flexibility, and countless other factors can be leveraged in sophisticated algorithms, exceeding human capabilities by far. The second cluster contains AI applications augmenting labor and capital. AI is increasingly capable of taking over time-consuming tasks with low value added, e.g., data research tasks, skimming unstructured data, and pre-selection of documents. Thus, workers can allocate more of their labor time to high-value work. Another example of AI-based profit optimization is predictive maintenance. AI can expand condition monitoring to scales unreachable for humans and even simulate systemic forces to flag potential problems. Thirdly, a cluster of AI applications is around increasing effectiveness in innovation. AI helps to identify promising avenues of innovation and reduces resources spent on unqualifying ideas. AI can drastically reduce the investments in trial-and-error phases of innovation and select promising ideas from the innovation funnel (Plastino & Purdy, 2018).

Further applications include, but are not limited to, process planning and monitoring, quality control, robotics, logistics, resource consumption optimization, energy management, and production interdependencies.

The recently fastened development of AI- and ML-based methods and tools for engineering and managerial problem-solving (accompanied by some vulgarization) caused terminological

mismatching, as well as misunderstandings (Kühl et al., 2022) that can potentially lead to low perceived trustworthiness of AI (Thiebes et al., 2021). This situation also motivates the clear, systemic analysis of the underlying methods and applications.

3.1.4 AI applications for sustainable manufacturing – the connection between AI and sustainability

The connection between AI and sustainability is multifaceted and evolving, offering significant opportunities. Three dimensions of this connection shall be examined: 1. The general interconnection between AI and sustainability, 2. AI's role in enabling sustainability within corporate environments, and 3. The application of AI in manufacturing to advance sustainability goals. Concrete examples from industry practice illustrate the practical implications of AI-driven sustainability.

AI, as a transformative technology paradigm, enables the results-driven processing of large datasets, identifying patterns, and allowing forecasting with unprecedented precision. AI can be used to utilize existing assets more effectively, enhance how data and facts are supervised and shared across various dimensions (Kar et al., 2022).

Khakurel et al. explored the impacts of AI in five sustainability categories – individual, social, economic, technical, and environmental. While impactful, AI is a tool and as such is up to the developers and stakeholders involved to take sound ethical decisions based on values commonly shared amongst citizens for the joint vision of a sustainable and resilient future (Khakurel et al., 2018).

One of AI's major contributions lies in human skill augmentation – supporting decision-making, interpreting complex data, pattern recognition, and predictive analysis. While AI struggles with intuitive reasoning and transferring insights into actionable solutions autonomously, its value multiplies when it is combined with human expertise.

Kar et al. found that multiple contemporary AI algorithms are highly used methods to improve the environmental, economic, and social pillars of sustainability. Due to the fast-paced development of methods and applications on the one hand, and the time requirement of sustainable management practices and their incorporation on the other hand, assessments remain in early stages (Kar et al., 2022).

In corporate environments, sustainability is both driven and challenged by diverse and often competing stakeholder interests. Shareholders value profits, employees desire purpose and ethical governance, regulators enforce compliance and accountability, and customers value a

product's balance of quality, value, price, and sustainability, to name but a few stakeholder interests that challenge corporate sustainability. AI does help navigate the stakeholder balance. In a recent study (P. Chen et al., 2024), researchers were able to measure the relationship between AI and corporate sustainable development, identifying three key impacts of AI in corporate environments:

- AI facilitates corporate sustainable development, with a more significant impact on environmental governance and social responsibility.
- Culture and public pressure impede the positive effects of AI, while technical proficiency and data security awareness amplify its positive impact.
- AI achieves corporate sustainable development by alleviating financing constraints, reducing agency costs, enhancing supply chain performance, labor productivity, resource utilization efficiency, and mitigating enterprise risks, among other factors.

Accelerating the innovation funnel, supporting (product) design for sustainability, improved data analysis for life cycle assessments, quality assurance enhancement, optimizing supply chains, there are rarely any corporate tasks where AI application is not optimizing performance.

In manufacturing industries, sustainability efforts often emphasize environmental sustainability. While social sustainability and governance are not neglected, resources tend to be allocated towards environmental sustainability. Sustainable manufacturing deep dives into resource and energy efficiency and thereby waste avoidance and pollution control.

Example energy efficiency – AI-enabled pattern recognition in energy use data allows demand-specific, efficient operation of production equipment. Production timing, volume, and maintenance intervals are optimized along the patterns of energy usage, product demand, and asset availability. In integrated productions, multiple different production operations locally connected and potentially interdependent, AI-algorithmic pattern recognition can optimize the interconnected production operations. For example, a steam and electricity-driven production with owned steam and power production assets could manage the production to leverage cost-optimized timing based on spot market energy prices, customer demand forecasting, raw material availability, and maintenance requirements.

Example quality control – today's velocity and precision of visual quality inspection is only possible with computer-driven imaging and image recognition. AI, in this case ML, allows for the quality control systems to become ever more accurate. This accuracy allows for the identification and elimination of production flaws, leading to higher output of quality products and thus less production waste. The quality control example extends to other manufacturing

tasks. Robotics and automation technologies, powered by AI, can optimize production processes, reduce energy consumption, and minimize waste generation (Olawade et al., 2024).

AI plays a pivotal role in the pursuit of sustainable manufacturing, supporting a wide range of objectives across environmental, social, and economic dimensions. From optimizing production systems and supply chains to enhancing stakeholder management and corporate governance, AI acts as a lever for sustainable transformation. As AI technologies continue to evolve, their alignment with long-term sustainability goals must be continually assessed, ensuring that progress in innovation translates into real-world impact.

3.2 Systematic literature review to establish a knowledge base

To support the systematic categorization and evaluation of conventional and AI/ML applications in manufacturing, a proper assessment of scientific literature on the topic is essential. Following the specific aims of the objective, acquiring knowledge of the current state of research on computer-based problem-solving in manufacturing, including conventional, AI, and ML applications, is the aim of the literature review. Ideally, a literature review delivers (1) both quantitative and qualitative synthesis of relevant research, (2) a comprehensive compilation of the pertinent scientific contributions, and (3) a systematic identification of computational problem-solving methods applied in manufacturing.

3.2.1 Search term and database selection

An initial search in three databases shows high numbers of findings (Table 4). Further refinement is required to conduct a meaningful and feasible systematic literature review.

Table 4 Initial search results in high numbers of findings

Search term (timeframe 2010 – 2025)	SCOPUS results	Web of Science results	Google Scholar results
computational AND problem-solving AND manufacturing	267	89	22100
artificial AND intelligence AND application AND manufacturing	4685	5064	25700
machine AND learning AND application AND manufacturing	5682	5918	17800

The search term needs to include all differentiators required to achieve the objective and allow a specific search, yielding most relevant results. For this review, after testing several variations, the following search term is utilized:

((compu* {problem solving}) OR (appli* "artificial intelligence") OR (appli* "machine learning")) AND manufacturing

If the database offered an advanced search option or a search-in-results functionality, an additional search was conducted using:

assess* OR review OR evaluat* OR compar*

Depending on the database's operators, the search term was adapted to produce the most relevant findings.

The selection of databases aims to ensure the inclusion of relevant English-language research while being reasonably feasible. An internet search for relevant and topic-appropriate databases, combined with the experiences of the author, was started with six databases: SCOPUS, Research Gate, Google Scholar, Web of Science, IEEE explore, and JSTOR. Research Gate was deselected because it does not present the number of findings, thus not allowing a comparison with other databases. Google Scholar was deselected because despite the use of further search refinement, the number of findings remained at 18,200, an unfeasible number of results. The four remaining database searches surfaced the following number of results shown in table 5:

Table 5 Results by database

Database	SCOPUS	Web of Science	IEEE explore	JSTOR
Number of results	534	4,274	146	229

3.2.2 Inclusion/exclusion criteria

To increase the relevance of findings while reducing the number of findings, a set of inclusion and exclusion criteria has been developed, listed in table 6 below.

Table 6 Inclusion/exclusion criteria

Criterion	Include	Exclude
Language	English	Non-English
Publication type	Published, peer-reviewed journal articles, book chapters	All else, incl. unpublished research, non-peer-reviewed sources, white papers, conference proceedings
Time frame	2015 – 2025 (February 28, 2025)	Findings published prior 2015
Intervention	Studies applying computational problem-solving in manufacturing	Purely theoretical studies; Focus does not include manufacturing

Outcome measures	Studies reporting quantitative/qualitative results related to the research question	Studies without measurable or clearly reported results
------------------	---	--

3.2.3 Screening

Applied to the results of the above search strategy, the study selection process begins with an initial screening of titles and abstracts. Given the number of findings, abstracts will only be screened if the title indicates relevance for the research. Relevance indicators are the use of the search terms in a context of applied computational problem-solving in manufacturing.

SCOPUS initial screening: 170 of the 534 findings are relevant based on the criteria described above. Ordered by relevance, the first 250 results included the majority of relevant studies. Search results 251-534 surfaced with less than a dozen relevant studies. A significant portion of relevant and irrelevant search results focus on additive manufacturing.

Web of science initial screening: the 4,309 findings of the Web of Science search pose a quantitative challenge for the initial screening. Since findings are ordered by relevance, a significant drop in the quota of relevant studies was experienced after 150 findings. Consequently, the screening included the first 250 findings, of which 157 results were deemed relevant. Interestingly, the Web of Science results include a significant focus on additive manufacturing.

IEEE explore initial screening: out of 147 findings, 64 results passed the initial screening. Those findings excluded in the initial screening were either purely theoretical or related to a different business sector, e.g., health, consumer applications, or society. This engineering-heavy database was a significant source of relevant studies.

JSTOR initial screening: both the 359 and the further filtered 229 results include few relevant findings. Despite the detailed search, the findings' focus topics are mostly outside manufacturing. Wind turbine efficiency, military engineering, STEM education are examples of the findings' topics. Only seven findings have been ingested into the Mendeley database.

In total, 1161 findings were ingested in a Mendeley reference manager software. After eliminating duplicates and entries lacking authors, the number of studies is 1007. Screening against the inclusion criteria eliminated 722 studies. Of the 285 full text articles in the database,

269 were retrieved, 16 full text articles could not be retrieved. Finally, 269 studies were included in the review.

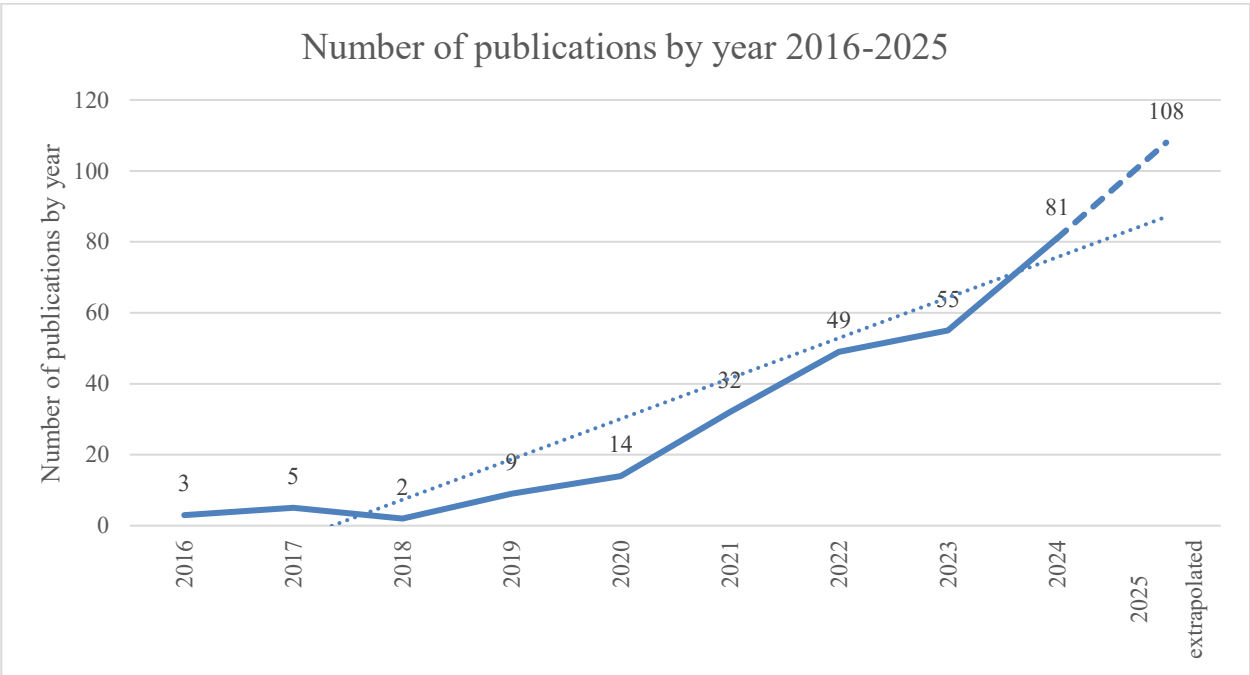


Figure 1 Number of publications by year (2025 value extrapolated from February 28 value 18: (18/2 months) x 12 months)

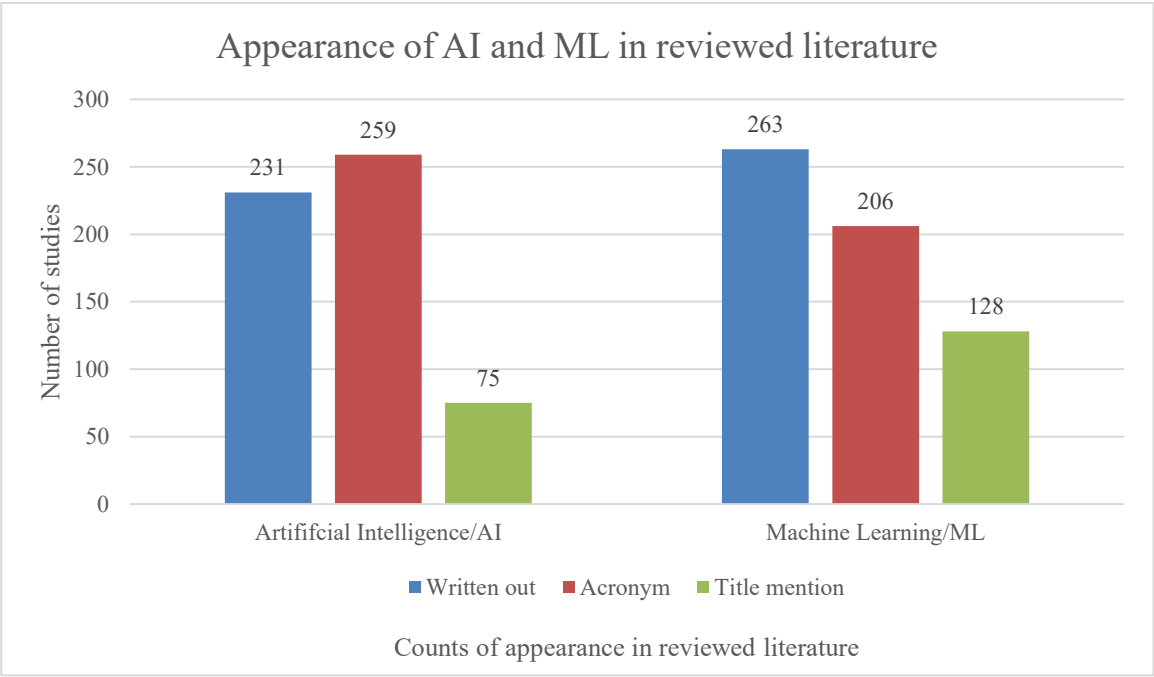


Figure 2 Appearance of AI and ML in reviewed literature

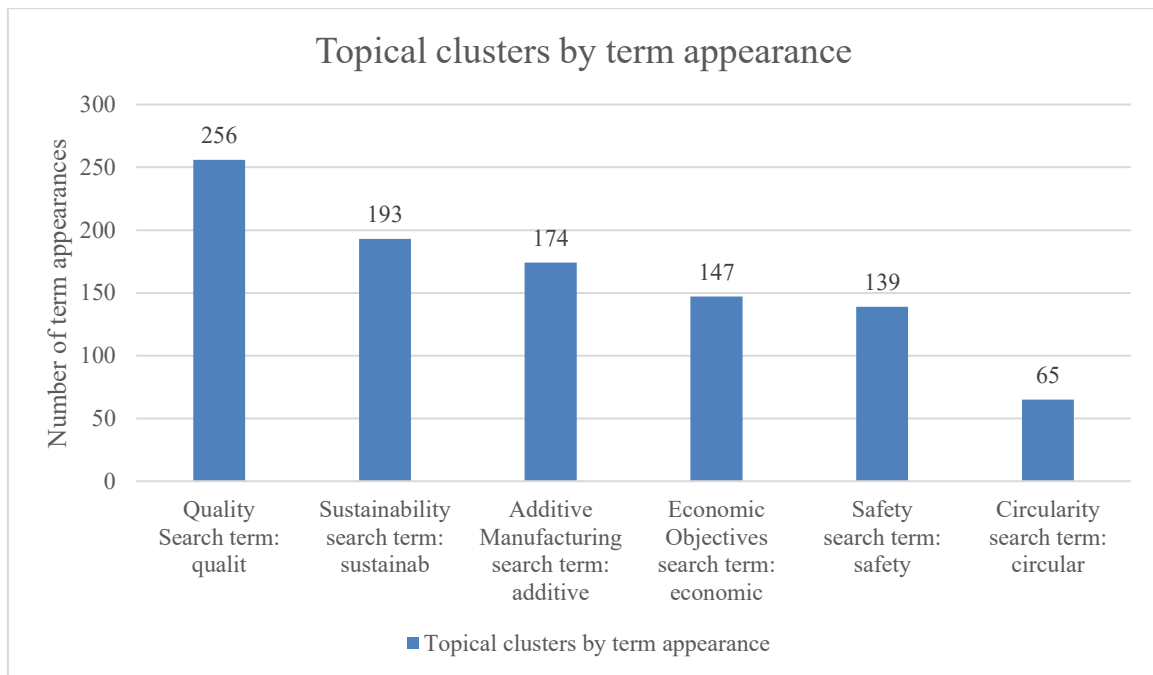


Figure 3 Topical clusters by count of search term appearance in reviewed literature

3.2.4 Topical cluster identification

The literature review surfaced multiple topical clusters. The most relevant clusters are:

- Quality
- Sustainability
- Additive Manufacturing
- Economic Objectives
- Safety
- Circularity

The quality dimension is mentioned in 256 out of 285 studies. Yet only ten titles include the term quality, indicating that the focus of most studies is not on quality. Sustainability is touched by the vast majority of studies, 193 out of 285. In the case of sustainability as well, only ten title mentions indicate other focal points. Additive Manufacturing, the third largest topical cluster in the review set (174 of 285), features 55 studies mentioning additive manufacturing in the title. An interesting find, as additive manufacturing is merely one specific form of manufacturing among many others. The extraordinary number of studies focusing on additive manufacturing reveals an interesting research subject: How are AI/ML and additive manufacturing connected? Is there an especially good fit of AI/ML and additive manufacturing compared to other manufacturing approaches?

More than half of the studies include a variety of economic objectives. Improved management and decision-making, process optimization, material flows, output increase, and quality improvement are among the sub-topics of the topical cluster economic objectives. Not surprisingly, in an industry that has worked on optimizing safety for decades, 139 of 285 studies touch on safety. None of the studies in the review mentions safety in their title, though. Finally, the sustainability-related topic of circularity is mentioned in 65 of 285 studies.

Assessing the six identified topical clusters, there are different levels of relevance for the research objectives among them. Sustainability, economic objectives, safety, and circularity are topics subsuming a broad range of subtopics. They can be considered meta-topics, all four highly relevant to answering the research questions of this dissertation. These four meta-topics will be the properties in property class four, “Development Goals.”

Sustainability

A true meta-term, sustainability subsumes a broad range of topics, leading to a multitude of sustainability definitions. Moore et. al. have identified five sustainability constructs and developed a comprehensive definition of sustainability based on over 200 sustainability studies including 24 definitions: After a defined period of time, a program continues to be delivered and/or individual behavior is maintained; the program and individual behavior change may evolve or adapt while continuing to produce benefits for individuals/systems. (Moore et al., 2017)

Scientifically derived, Moore et. al.’s definition does not explicitly include the context of our whole planet, the integrity and health of its biosphere, and the future well-being of humanity. Ben-Eli captures the various dimensions of sustainability by first identifying five domains of sustainability:

- The Material Domain, which constitutes the basis for regulating the flow of materials and energy that underlie existence.
- The Economic Domain, which provides a guiding framework for defining, creating, and managing wealth.
- The Domain of Life, which provides the basis for appropriate behavior in the biosphere in relation to all other species.
- The Social Domain, which provides the basis for social interactions.
- And, the Spiritual Domain, which identifies the necessary attitudinal, value orientation and provides the basis for a universal code of ethics. (Ben-Eli, 2018)

Another broadly used approach to define sustainability and sustainable manufacturing is the so-called Triple Bottom Line (TBL), classifying the concept of sustainability into three dimensions economic, environmental, and social. The environmental dimension includes emissions, pollution, resource consumption, and biodiversity. The economic dimension breaks down into profit, manufacturing costs, and investment. Lastly, the social dimension includes employees, customers, and community. (Eslami et al., 2019) The TBL can be broken down into further detail; for this study, the level of detail is sufficient.

The three presented definitions of sustainability, when integrated, provide a sufficiently clear understanding of the concept of sustainability. Manufacturing companies balance the different dimensions of sustainability with differing effort levels depending on the business advantage at hand. In cases where sustainable practices come together with economic gain, e.g., efficiency improvements, resource or waste reduction, reputation, and recruitment, companies are willingly making changes towards more sustainable practices. Sustainable practices that come without immediate economic advantage or even disadvantage have a steeper threshold for execution.

Economic objectives

Formally ending a shareholder-only perspective, the Business Roundtable, an association of more than 200 chief executive officers of America's leading companies, issued a statement on the purpose of the corporation in August 2019. Generating long-term value for shareholders remained in the definition of business purpose, but, reflecting modern realities, the corporate purpose was broadened to include upstream and downstream stakeholders, employees, and communities. (Business Roundtable, 2019)

Economic objectives of manufacturing enterprises aim at ensuring profitability, growth, and competitiveness in the production of goods. Profit is maximized by optimized resource utilization, minimized costs, and constantly improved productivity. Reducing production costs through lean manufacturing, automation, and just-in-time inventory systems is crucial. Efficient use of raw materials, labor, and energy contributes to overall cost savings. (Ebert & Griffin, 2016 or any other business essentials book used in academic introductory courses) To survive and grow in dynamic markets, manufacturers must offer products that are competitive in terms of price, quality, and innovation. (Michael E. Porter, 1998)

While quality, as well, is a broad topic that can be broken down into various subtopics, (product) quality is immediately connected to the economic success of manufacturing companies.

Meeting or exceeding the (customer) expectations including previously set standards and specifications, as well as the absence of defects ensuring proper function, is a cornerstone of economic success for manufacturers, the competitive advantage Porter defined. Therefore, quality can be identified as one sub-topic of the cluster of economic objectives.

Safety

For decades, safety has been a top priority in manufacturing. In the manufacturing industry, the term safety includes all practices, protocols, and systems aiming to protect employees, equipment, and the environment from accidents or harm. The larger topic is structured into workplace safety, machine safety, process safety, and safety culture, among others. Ultimately, safety in manufacturing reduces risks, avoids downtime, enhances productivity, and protects humans.

There is an abundance of scientific and non-scientific literature on safety in general and safety in manufacturing in particular. Recent literature, published after 2010, is either a new edition or covers a specific sub-topic like personal protective equipment or machine guarding. Recent general overviews are provided by government authorities or supranational institutions, defining safety standards backed by scientific research. For example, the U.S. American Occupational Safety and Health Administration (OSHA) offers a General Industry Digest. (Occupational Safety and Health Administration (OSHA), 2015) The International Organization for Standardization publishes the ISO 45001 Standard on occupational health and safety management systems (ISO, 2018), and the National Safety Council provides accident prevention guidelines for industrial operations. (National Safety Council, 2015)

Two works found in safety managers' bookshelves are *The Handbook of Occupational Safety and Health* (Mansdorf, 2019) and *Todd Conklin's Pre-Accident Investigations* (Conklin, 2019).

Circularity

A rather recent topic for industrial manufacturing is circularity. A circular economy, while there are plenty definitions (Kirchherr et al., 2017), aims to decouple economic growth from resource consumption, contrasting the linear economy of take, make, and dispose. (Geissdoerfer et al., 2017) In a manufacturing context, circularity refers to a production model that emphasizes resource efficiency, waste minimization, and the strict reuse and recycling of materials to extend product lifecycles.

3.3 Literature in the context of how we can learn from the experts' opinions

Through learning and experience, human experts hold knowledge about all aspects of manufacturing. The knowledge accumulation started in pre-computational eras of manufacturing and continued through the stages of increasing use of computational problem-solving in manufacturing until today. Therefore, experts carry a valuable knowledge base for computational problem-solving.

Expert systems, or Knowledge-based systems, leverage human expert knowledge, coded into facts, rules, heuristics, and procedures, to resolve problems requiring significant human expertise for the solution. (Leo Kumar, 2019) Harnessing human knowledge and leveraging it for computational problem-solving is a topic of extensive scientific literature. Most recently, literature focuses on the critical integration of domain knowledge into AI/ML systems. The role of domain experts is evolving, and tools and technologies are emerging that help facilitate the incorporation of human expertise into AI/ML models. (Miller et al., 2024)

Trust in expert systems depends predominantly on the quality and dependability of the results and the explainability of the applied inference. Fuzzy models are easier to understand and useful when harnessing expert knowledge, gaining accuracy from the expert knowledge used to build them. (Wieland & Mirschel, 2017)

This dissertation acknowledges the distinction between explicit knowledge, which can be formally codified in rules and procedures, and tacit knowledge, which is experience-based, intuitive, and often resistant to formalization (Nonaka & Takeuchi, 2000). The challenge of embedding tacit knowledge into computational models underscores the importance of human-in-the-loop approaches, where experts remain actively engaged in model design, validation, and refinement (Dudley & Kristensson, 2018). These approaches not only enhance the reliability of computational models but also foster trust by maintaining human oversight in decision-making processes. Recent advances further demonstrate the value of hybrid systems that combine expert knowledge with data-driven machine learning, thereby benefiting from both structured domain reasoning and adaptive pattern recognition (Von Rueden et al., 2023). Such hybridization is increasingly recognized as a promising path for complex domains like manufacturing, where both human expertise and large-scale data hold indispensable value.

Given the complexity of manufacturing, experts may have differing views on specific problems. Beyond measurable and accurately calculable outcomes, differing opinions and alternative perspectives are common both in scientific literature and manufacturing reality. To harness

diverse expert assessments, expert evaluations allow incorporating multiple views in the expert system model. Asking different experts for their evaluation of a specific problem or their success prognosis, ideally double blind, allows to derive average evaluations or weighted average evaluations. Expert evaluation allows to validate the knowledge base, assess the accuracy of a model, tune and refine the model's rules. More generally speaking, the evaluation of machine learning algorithms is an important component of any research procedure that ascribes seriousness and credibility to the results (J. U. Hansen & Quinon, 2023).

3.4 Methods for discovery of methodological combinations

Considering the need outlined in the previous sections, various published methods can be found in the literature; among them are Bayesian Optimization (Yang et al., 2024), Active Learning (Tang & Huang, 2021), Reinforcement Learning for Method Selection (Torres et al., 2022; Vairavasundaram et al., 2023), Knowledge-Graph Reasoning (Zeng et al., 2023) and Genetic Algorithms (Petrowski & Ben-hamida, 2017).

A Bayesian network is a probabilistic approach that represents a set of variables and their conditional dependencies, using a directed acyclic graph. On this basis, Bayesian Optimization focuses on finding the global optimum, efficiently, while Bayesian updating revises beliefs as new evidence is available (Oijen, 2024). Bayesian Optimization might be applied efficiently for identifying and combining hyperparameters for better task-tool matching (Yang et al., 2024). Also, Hidden Markov Models, Markov Random Fields, or Decision Graphs (Sucar, 2021) can be mentioned among the probabilistic methods.

Active Learning is a subset of Machine Learning methods in which a learning algorithm can query users (i.e., experts) interactively to label data with the desired outputs (Tharwat & Schenck, 2023). It can be helpful tackling the challenge of hard-to-acquire and costly expert evaluation, as the algorithm proactively selects those subsets of examples to be labeled next from the pool of unlabeled data that are deemed to help the most if their labels are known.

Genetic Algorithms (GAs) are population-based metaheuristics, inspired by biological evolution, that evolve candidate solutions through selection, crossover, and mutation to solve optimization, search, or time-sensitive task allocation problems (W. Liu et al., 2025; F. Wang et al., 2024). GA's memory is limited to the population of some complete combinations, and learning is controlled by the evolving schemes.

According to the above overview, an actual challenge is that complex systems of sustainable manufacturing require sophisticated methodologies along with a clear systemic analysis of the

underlying complex tasks and of appropriate combinations of conventional and AI-based computational applications for the solution of these tasks. The literature, based on real-life use cases, provides exemplary assessments of utilizable methods and tools. However, a methodology is needed to learn from these solutions, as well as to suggest possible new combinations from the accumulated knowledge base.

Existing discovery approaches, as described in the overview above, are either probabilistic, optimization-driven, or population-based evolutionary. They do not provide a structured classification-based knowledge representation, combined with expert feedback. This identified gap motivates the methodological development in the following Chapter 4.

4. MATERIALS AND METHODS

4.1 Computational problem-solving methods involved in the further classification

This section provides an overview of the landscape of computational tools used in manufacturing, both conventional and AI/ML-based. The following set of computational problem-solving methods, systematically extracted from the scientific literature, establishes not only a consolidated reference point for the field but also the foundation for this dissertation's objective: to introduce and to test a methodology for the systemic categorization and evaluation of conventional and AI/ML applications in manufacturing.

4.1.1 Conventional computational problem-solving methods explained

The distinction between conventional and AI/ML-based methods used in this dissertation is conceptual, not chronological. The term “conventional methods” in this dissertation refers to approaches that are primarily model-based, rule-based, or analytically defined, and are not based on data-driven learning. AI/ML-based methods are characterized by their ability to adapt based on data, feedback, or training processes. In manufacturing, conventional computational problem-solving methods have long formed the foundation for decision-making, process optimization, and quality control. These methods are typically grounded in mathematical modeling, operations research, control theory, and statistical analysis. They offer deterministic approaches to solving complex problems in production planning, scheduling, inventory management, and process control. Despite recent advancements in AI, these traditional methods remain highly relevant due to their interpretability, stability, and underlying, well-established research. And they often serve as the basis for hybrid solutions when integrated with intelligent systems. Understanding these established approaches is essential for building effective problem-solving frameworks for addressing today's manufacturing challenges.

The typical set of conventional computational problem-solving methods in manufacturing, considered in this work, includes the following methods and tools: (Dynamic) simulation (Law, 2013); Model-based Predictor-Corrector (MPC) (Bertsekas, 2019); Statistical Process Control (SPC) (Montgomery, 2019); Mixed Integer Linear Programming (MILP) (Wolsey & Nemhauser, 2014); Mixed Integer Non-Linear Programming (MINLP) (Grossmann & Trespalacios, 2013); Finite Element Analysis (FEA) (Hughes, 2012); Data Mining and Analytics (Han et al., 2023); Computer-aided Design (CAD) (Encarnacao et al., 2012); Computer-aided Manufacturing (CAM) (Groover, 2020); Real-time Monitoring and Control

Systems (Buttazzo, 2011); Simulation-based Optimization (Fu, 2015); Expert Systems (Giarratano & Gary D. Riley, 2004).

(Dynamic) Simulation

Dynamic simulation models a system's evolution over time. In engineering, where systems are subject to change and complexity, dynamic simulation proves valuable. Typically, a dynamic simulation starts with an abstract representation of a system, including variables and equations to describe the dynamics. The simulation then uses numerical methods to solve the system's differential or difference equations (Law, 2013).

Model-based Predictor-Corrector (MPC)

MPC is an advanced computational method to solve complex (optionally optimal) control problems, integrating predictive modeling with corrective algorithms to refine solutions towards an optimal state. MPC problem-solving starts with a predictive model that estimates the outcome of a given set of variables. It provides a forecast of the system's behavior under various scenarios, allowing to navigate the search for optimal solutions. The corrector component is activated once a prediction is made. This iteration of prediction and correction continues until success criteria are met (Boyd et al., 2010; Bertsekas, 2019).

Statistical Process Control (SPC)

SPC focuses on monitoring and controlling processes through data analysis. Other than deterministic methods, it is handling uncertainty and randomness specifically well. Statistical tools are applied to identify and reduce variability in manufacturing processes. Due to its ability to assess process abilities to produce specified output parameters, it is applied in industries to improve quality and efficiency. SPC aligns with Six Sigma and Total Quality Management (Breyfogle, 2003; Montgomery, 2019).

Mixed Integer Linear Programming (MILP)

MILP solves complex optimization problems for (simplified) linear models with variables constrained to integer values exclusively. MILP is therefore useful in applications with discrete decisions. The combinatorial nature of integer constraints requires sophisticated algorithms for efficient solution finding. Applied in broad systems optimizations, MILP shows extraordinary versatility in handling real-world problems (Wolsey & Nemhauser, 2014).

Mixed Integer Non-Linear Programming (MINLP)

Nonlinearity and mixed-integer variables make MINLP problems challenging. MINLP is distinguished by combining integer programming and nonlinear programming to solve complex problems. A MINLP problem starts with an objective function and constraints, providing a framework to model and solve real-world problems involving discrete choices and nonlinear relationships. This problem-solving method is broadly used in engineering design, energy systems, chemical process design, and supply chain management, among other use cases (Quesada & Grossmann, 1992; Leyffer, 2001; Grossmann & Trespacios, 2013).

Finite Element Analysis (FEA)

Finite Element Analysis approximates the behavior of complex systems under multiple physical forces. Engineering, physics, and applied sciences apply this numerical method for solving problems including geometries and boundary conditions. FEA divides problems into more manageable parts called finite elements. Models then assemble these elements, describe each element's behavior, and approximate the system's behavior (Hughes, 2012).

Applied in various industries, FEA is used for stress tests, thermal and vibration analysis, and fluid dynamics. Stress distribution and deformation analyses are common results achieved using FEA algorithms (Gockenbach, 2006).

Data Mining and Analytics

Data Mining is the process of identifying patterns within large datasets. By using different techniques, the aim is to extract meaningful information from data. By combining statistics, computer science, and database technology, this interdisciplinary field infers patterns, correlations, anomalies, and trends in data that are not immediately apparent (Han et al., 2023). Analytics interpret the findings of data mining and apply it for decision making. Descriptive analytics look back, predictive analytics forecast the future, and prescriptive analytics propose actions. Analytics leverage data-driven insights to improve processes and efficiency (Davenport et al., 2010). Data mining and analytics have become a competitive differentiator in the digital age.

Computer-aided Design (CAD)

CAD uses computer systems to support the creation, modification, analysis, and optimization of a design. CAD provides an input tool to graphically represent physical dimensions and relationships of objects. CAD also offers to simulate behaviors of designs under specific conditions, allowing them to confirm feasibility of designs before physical production. CAD

systems support both two-dimensional drafting and three-dimensional modeling. The integration of design and manufacturing processes optimizes workflows and reduces errors (Encarnacao et al., 2012; Groover, 2020).

Computer-aided Manufacturing (CAM)

CAM uses software and software-controlled hardware to automate manufacturing. Often integrated with CAD systems, CAM may cover the complete manufacturing process from design to the final product. Objectives of CAM are efficiency enhancements, quality improvements, waste reduction, and production flexibility. CAM allows manufacturing beyond the limits of manual processes. Capable of multi-axis machining, CAM can facilitate the production of otherwise impossible geometries (Groover, 2020).

CAM has allowed formerly unseen precision levels and tight tolerances in manufacturing. It facilitates modern methodologies like just-in-time, just-in-sequence, lean manufacturing, and mass customization (Sharma et al., 2023).

Real-time Monitoring and Control Systems

Real-time monitoring and control systems are designed to manage processes by deriving actionable conclusions from continually collected and analyzed data. In domains where timely decision-making is critical, these systems ensure efficiency, safety, and reliability. The real-time monitoring part of the system continually gathers sensor data and provides an always current state of the monitored system. The control part of the system leverages these insights to steer actuators or control mechanisms toward a desired outcome, e.g., optimal performance or risk mitigation (Liu, 2000; Buttazzo, 2011).

Simulation-based Optimization (SBO)

SBO combines simulation modeling with optimization techniques to solve complex problems where traditional methods are not successful. Applied widely in several domains, SBO is particularly successful in dynamic stochastic systems too complex for closed-form equations. SBO generates a simulation model of a system to test and evaluate performance in a virtual setting, allowing to experiment on the virtual system twins under controlled conditions. The optimization part of SBO searches the given solution space to propose the most advantageous settings for a specific objective function (Fu, 2015).

SBO can explore wide ranges of scenarios, allowing to assess scenarios that are hard or impossible to test in reality. With the recent advancements of computational power and

algorithms, SBO is central to modern decision making in engineering and manufacturing (April et al., 2003).

Expert Systems

Expert systems are emulating human experts' decision-making. Expert systems approach complex problems with reasoning through bodies of knowledge, using if-then rules rather than procedural code. At the center of expert systems is an inference engine that applies logical rules to derive information or decision proposals. The system does not execute its solutions but proposes them to a human operator. Expert systems can explain their reasoning and can admit their limitations. Modern expert systems update their knowledge base to improve their performance over time (Jackson, 1998; Giarratano & Gary D. Riley, 2004).

4.1.2 AI/ML-based computational problem-solving methods explained

AI/ML-based computational problem-solving methods have introduced a paradigm shift in manufacturing by enabling adaptive, data-driven, and predictive decision-making. Unlike conventional approaches, these methods excel in handling high-dimensional, non-linear, and uncertain data environments common in modern production systems. These tools offer the ability to learn from historical and real-time data, uncover hidden patterns, and continuously improve performance without explicit programming. While their interpretability and integration with legacy systems remain ongoing challenges, AI/ML methods provide significant advantages in scalability, responsiveness, and performance in complex environments.

Leveraged by themselves or to augment conventional computational problem-solving methods, the following typical AI/ML-based computational problem-solving methods and tools are used in manufacturing: Convolutional Neural Networks (Goodfellow et al., 2016), Support Vector Machines (Cortes & Vapnik, 2020), Random Forests (Ho, 1995), k-Nearest Neighbors (Alpaydin, 2020), Fuzzy Logic and Reasoning (Novák, 2012), Genetic Algorithms (Mitchell, 1998), and Evolutionary Learning Strategies (Beyer & Sendhoff, 2008).

Convolutional Neural Networks (CNNs)

A CNN (or ConvNet) is a deep learning architecture that uses the concept of convolution to learn features that describe visual input (Goodfellow et al., 2016). They can be trained for various tasks, such as classification, detection, localization, and segmentation. The structure of a CNN is described by Russakovsky et al. (Russakovsky et al., 2015) as an input layer, convolutional layers, pooling layers, and fully connected layers. The input layer receives the raw input data, such as an image, and then passes it to the first convolutional layer. As stated

by Goodfellow et al., each convolutional layer applies a set of learnable filters to the input data to generate feature maps that represent the activations of learned features across the input data. The pooling layers then subsample the feature maps to reduce their dimensionality (Goodfellow et al., 2016). The output of the final pooling layer is then passed to one or more fully connected layers, which aim to perform high-level reasoning (Gu et al., 2018).

Support Vector Machines (SVMs)

A support vector machine (SVM) is a supervised learning algorithm (Cortes & Vapnik, 2020). It is a classification model that finds the hyperplane that separates two data classes while maximizing the distance between them. This hyperplane is a flat geometric object containing data points belonging to one class but not to any other class. To create this hyperplane, SVMs make use of a training dataset (or "training set") that has been labeled with the desired outputs. The SVM algorithm learns from these training examples to find the hyperplane that separates the closest examples of two classes from each other in a multidimensional feature space (Chang & Lin, 2011). Once the hyperplane is found, it can be used to classify new data points by determining which side of the hyperplane they fall on. SVMs have been used successfully in various applications, including image recognition, text classification, and bioinformatics. One advantage of SVMs is that they are less prone to overfitting than other machine learning algorithms, as they rely on finding the optimal hyperplane that separates the data points with the largest possible margin rather than trying to fit a complex model to the data. However, SVMs can be computationally intensive, especially when dealing with large datasets or non-linear problems.

Random Forests

Random forests are an ensemble model in machine learning and statistics that combines multiple decision trees. The ensemble model is created by bootstrapping and aggregating decision trees. The idea of random forests was first introduced by Tin Kam Ho in 1995 (Ho, 1995), but the concept was further developed by Leo Breiman and Adele Cutler (Breiman, 2001). The decision trees in a random forest are trained on a random subset of the training data and a random subset of the input features. By introducing randomness in this way, the algorithm can prevent overfitting and reduce the variance of the model. The best feature to split on is determined by maximizing information gain. This recursive splitting of the data is continued until the leaf nodes are reached, which contain a predicted label. The final prediction of the random forest is obtained by taking the average of the predicted labels from all the trees in the forest. (Breiman, 2001). Random forests have several advantages over other machine learning

algorithms, such as linear regression and support vector machines. Random forests have a higher tolerance for noisy data and can handle missing values better. In addition, random forests do not rely on any assumptions about functional form or location to generate predictions.

k-Nearest Neighbors (k-NN)

K-Nearest Neighbors (k-NN) is a popular ML algorithm that leverages the concept of proximity to identify similar patterns in data (Alpaydin, 2020). Its effectiveness in predicting class membership has made it a preferred choice in various industries, including manufacturing, for applications such as quality control and process monitoring (Tobin et al., 1998). The algorithm can be used in manufacturing and quality control applications where prior knowledge of the data is not available (Cavalcante et al., 2019). The k-NN algorithm works through a series of steps. First, a training set is used, which consists of examples that contain values of the attributes being analyzed and their corresponding class membership (Alpaydin, 2020). Attributes refer to specific characteristics of data points on a graph, such as the function of an equipment or its color. Next, the algorithm assigns known class memberships to the k closest points in the data set. This is why it's called a "k" nearest neighbors algorithm - only the closest points are used to predict an unknown point (Tobin et al., 1998). The predicted class membership for an unknown point is then compared with its true class membership. If they are similar enough, the class membership for the unknown point is identified. Otherwise, adjustments to the model or a different value for "k" may be necessary to improve the accuracy of the predictions (Alpaydin, 2020). The algorithm's effectiveness lies in the idea that similar patterns are located close to each other, making it a powerful tool for identifying patterns in data and predicting class membership (Alfere & Maghari, 2018).

Fuzzy Logic and Reasoning

Fuzzy Set Theory (Zadeh, 1965) is based on the extended definition for the membership of set elements to a subset, by $[0, 1]$ real values. The respective Fuzzy Logic is a mathematical method for representing vagueness and allowing for partial truths (Novák, 2012). Fuzzy logic systems are a type of AI that makes decisions based on fuzzy set theory, rather than a traditional, clear-cut approach (Ross, 2009). These systems use two logical variables, membership functions, and weights, to determine how fuzzy the system is (Tanaka et al., 1998). Membership function determines which values should be accepted or rejected by the system, while the weights influence the system's decisions (Ross, 2009). Fuzzy logic is often used in industrial manufacturing to make decisions based on incomplete or imprecise information (Dadios, 2012). It is especially useful when manufacturing systems are designed with specific tolerance levels

for parts and materials (F. C. Chen et al., 2010). With fuzzy logic, these decisions can be made more accurately, reducing waste and increasing value (Oblak et al., 2017). Fuzzy logic systems have drawbacks, such as requiring extensive planning during their development stages and uncertainty about which system is best for a specific application due to their vast number of uses and applications.

Genetic Algorithms

A specific class of evolutionary computation techniques initiated in the 1970s, Genetic Algorithms (GAs) simulate the process of natural selection. Beginning with a population of candidate solutions represented by a chromosome-like data structure in the form of binary values (Goldberg, 2013), GAs mimic biological genes to encode possible solutions to an optimization problem. Three operators characterize GAs. Selection based on fitness, crossover or recombination to produce new offspring, and mutation to provide genetic diversity to explore the edges of the solution space. (Jong, 2017) Particularly known for their adaptability and capability in complex solution spaces, GAs became a popular choice in science and industry (Mitchell, 1998).

Evolutionary Learning Strategies

In science and engineering, Evolutionary Learning Strategies (ELS) have proven to be particularly effective for the optimization of high-dimensional, complex problems. ELS evolved the binary encoding of traditional genetic algorithms to real-valued vectors. This made ELS particularly suitable for continuous optimization tasks in real-world scenarios. (Beyer & Sendhoff, 2008) An ELS starts with a population of candidate solutions, represented as real numbers. The candidates are assessed by a fitness function for the specific optimization problem. The candidates are mutated by random small changes and recombination of elements from different candidates (N. Hansen et al., 2003).

4.2 The method of Reevaluable Possibility Space: an equivalence classification and evaluation feedback-based learning

Considering the dissertation's objectives and specific aims, the following features must be taken into consideration in the development and application of a new method for the discovery of methodological combinations:

- The constructive equivalence classification of underlying properties of the tasks and the applicable methodologies or methodological combinations;

- Commutative and associative character of (compatibility) relations between the properties within the combinations (in graph representation, it is described by non-directed edges);
- Rather a possibilistic than probabilistic kind of associated uncertainty (In discovery, the value of a new solution does not depend on its frequency, similarly to innovation);
- Consistent representation of ‘lack of knowledge’;
- Expressing of some essential properties of evaluation distributions, coming from expert’ suggested heuristic values;
- Algorithmic generation of aggregation and optionally the consent of evaluations from multiple experts.

Accordingly, the applied methodology is based on the iterative reevaluation of the lattice of the compatible properties, classified into equivalence classes. A lattice is an abstract structure studied in the mathematical subdisciplines of order theory and abstract algebra (Grätzer, 2003). The special compatibility lattice of the discrete properties was introduced by Blickle (Blickle, 1977). This lattice belongs to a family of optionally fuzzy valuated relational algebras (Arva & Csukas, 1989). An example was published in (Csukás et al., 2012), where the methodology was applied for supporting polymer composite experimentation.

4.2.1 Minimal and maximal combinations in a property lattice

The formal description of the lattice, as relational algebra (Arva & Csukas, 1989), is given by the quintuple

$$\langle S, \emptyset, \omega, \overline{\cup}, \underline{\cap} \rangle \quad (1)$$

where S is the set of elements in the structure; $\emptyset$ and ω are special elements; while $\overline{\cup}$ and $\underline{\cap}$ are the connectives, composing and decomposing the combinations, according to the definitions of

$$s_i \overline{\cup} s_j = \begin{cases} s_k = s_i \cup s_j & \text{iff } s_k \in S \\ \omega & \text{otherwise} \end{cases} \quad (2)$$

and

$$s_i \underline{\cap} s_j = \begin{cases} s_k = s_i \cap s_j & \text{iff } s_k \in S \\ \emptyset & \text{otherwise} \end{cases}, \quad (3)$$

respectively. All the elements are given as a subset from the power set of the minimal (generating) elements $\check{S}$, while in the set S contains the maximal elements $\hat{S}$, as follows:

$$\hat{S} \subset S \subset P(\check{S}) \quad (4)$$

The minimal and maximal combinations are defined by the following expressions:

$$\forall_j \quad \check{s}_1 \sqcap s_j = \emptyset \quad \text{and} \quad \forall_j \quad s_j \sqcap \check{s}_1 = \emptyset \quad (5)$$

$$\forall_i \quad \hat{s}_k \sqcup s_i = \omega \quad \text{and} \quad \forall_i \quad s_i \sqcup \hat{s}_k = \omega \quad (6)$$

The v order of elements $s_i^v \in S$ determines the number of the minimal (generating) elements in the given elements.

Specially, the compatibility lattice (Blickle, 1977) is built from properties T , classified into equivalence classes T_i

$$T = T_1 \cup T_2 \cup \dots \cup T_n \quad (7)$$

while the generating minimal elements $\check{S}$ are compatible pairs of properties

$$\check{S} \subset T \times T \quad (8)$$

belonging to two different equivalence classes, and the maximal combinations are property sets containing one and only one property from each equivalence class.

4.2.2 Evaluation of the property lattice

The lattice structure (Equation (1)) can be extended to an evaluated structure (Arva & Csukas, 1989; Csukas et al., 1989), as follows:

$$\langle S, \emptyset, \omega, \overline{\cup}, \sqcap, m, M, \max, \min \rangle \quad (9)$$

$$\hat{S} \subset S \subset P(\check{S}) \quad (10)$$

$$\hat{s}_i \rightarrow V(\hat{s}_i) \in [0,1] \quad (11)$$

$$s_j \rightarrow m(s_j) \in [0,1] \quad (12)$$

$$s_j \rightarrow M(s_j) \in [0,1] \quad (13)$$

$$m(s_k \sqcap s_1) = \min[m(s_k), m(s_1)] \quad (14)$$

$$M(s_k \sqcap s_1) = \max[M(s_k), M(s_1)] \quad (15)$$

$$m(s_k \bar{\cup} s_1) = \max[m(s_k), m(s_1)] \quad (16)$$

$$M(s_k \bar{\cup} s_1) = \min[M(s_k), M(s_1)] \quad (17)$$

where the former quintuple (Equation (1)) is extended with the m and M normalized evaluations of the worst and best solutions containing the given subset, as well as with the minimum ($\min$) and maximum ($\max$) operators (Equation (9)). According to Equation (10), the set of combinations $\check{S}$ can be interpreted as a subset in the power set of minimal (binary) combinations, while the set of maximal complete combinations $\hat{S}$ is a subset of set S . In the resulting Reevaluable Possibility Space (RPS), $V(\hat{S})$, according to Equation (11), means the normalized evaluation of the maximal combinations, while $m()$ and $M()$ are for the lower and upper bounds of the uncertain evaluations of the lower-order (in the marginal case: minimal binary) elements in the sense of Equations (12) and (13).

In the learning phase of RPS, we determine the uncertain evaluation of the minimal relations by the intervals $[m, M]$, with the knowledge of the evaluation of the maximal combinations. In the discovery phase, considering these uncertainly evaluated binary relations, we can generate proposed solutions in the form of new maximal combinations. The change in the lower and upper bounds of the disjunctive $\sqcup$ and conjunctive $\bar{\cup}$ operators can be calculated by Equations (14) and (15), as well as Equations (16) and (17), respectively.

The RPS is also prepared for discrete (Yes/No, 0/1) evaluations of maximal elements (complete combinations). In this special case, the lower-order (finally binary) elements may have four values. In the respective four-valued logic, the elements may be true ($m=1$ and $M=1$), false ($m=0$ and $M=0$), uncertain ($m=0$ and $M=1$), and unknown ($m=1$ and $M=0$). The completeness of the valuated lattice (9) can be seen from the facts that special elements $\emptyset$ and ω are ‘uncertain’ and ‘unknown’, respectively.

According to Equation (11), the maximal elements (complete combinations) are characterized by an unambiguous evaluation that can be normalized into the $[0, 1]$ interval. The lower-order combinations, and finally, the minimal (binary) elements, have increasingly uncertain evaluations. It is to be noted that in a more sophisticated interpretation, the lower-order elements can be associated with a changing evaluation distribution that represents the distribution of the value of maximal elements they participated in. Considering the difficulties of algorithmizing these distributions, a simplified approach can be used, characterizing the uncertainty with the domain of these distributions, which can be defined by the $[m, M]$ sub-intervals of the whole

[0, 1] interval. Accordingly, the uncertainty (U) of minimal (binary) elements ($\check{S}$) can be calculated as follows:

$$U = \sum_{i=1}^N \text{abs}(M(\check{s}_j) - m(\check{s}_j)) \quad (18)$$

where

$$\check{s}_j \in \check{S} \quad (19)$$

and

$$N = \text{card}(\check{S}) \quad (20)$$

while uncertainty U can be normalized (resulting u) by dividing with the number N of binary elements:

$$u = \frac{U}{N} \quad (21)$$

In the case of lack of knowledge, uncertainty (U) is negative; that is why the uncertainty of ‘lack of knowledge’ and of the evaluation-based uncertain knowledge have to be distinguished. Nevertheless, their absolute values may be summarized.

4.2.3 Reevaluable Possibility Space for generation of problem-solving methods

Considering the huge number of possible scenarios on the one hand, as well as the heuristic knowledge of collaborating experts on the other, it is an effective way to generate the possible scenarios in the above-outlined RPS structure. In this methodology, the features (characteristics, properties, etc.) of analyzed solutions are described by a compatibility lattice (Equations (1)-(6)), consisting of feature equivalence classes (Equation (7)) and binary compatibility relations (Equation (8)) between the elements of the various classes.

In an equivalence classification, each feature belongs to one class exclusively, while the compatible maximal combinations ($\hat{S}$), built from binary relations ($\check{S}$), contain one and only one feature from each class. For example, the various tasks and levels of manufacturing systems, the various computational methods, etc., determine the feature equivalence classes, while compatibility of properties is declared by compatibility relations between them (e.g. between a given task or level and a given AI-based computational method, etc.).

The essential features of the uncertain evaluation feedback-based learning algorithm are the following:

- According to Equation (10), the complete combinations ($\hat{S}$) can be evaluated by one or more experts or objective functions, unambiguously, while
- The set of partial property combinations and ultimately the set of smallest binary relations ($\check{S}$) can take place in multiple solutions of different values, so their evaluation is inherently uncertain.

The appropriately described uncertain values $[m, M]$ of minimal (binary) relations contain an uncertain knowledge that can be used for the uncertain suggestions of new complete combinations.

The complete combinations ($\hat{S}$) can be evaluated by a single real value (V in Equation (11)), normalized by the temporarily achieved best value to the $[0, 1]$ interval. The partial combination of features (and finally the binary compatibility relations, $\check{S}$) inherits an uncertain knowledge, expressing the evaluation of maximal combinations they participated in. This knowledge can be utilized for the discovery of new candidates for the next maximal combinations to be evaluated. Optionally, the method can be extended for the consideration of opinion from multiple evaluating experts, while an automatic consent can be derived from time to time, and each objective follows the evaluations, according to the consensus-related knowledge base.

A considerable amount of ‘a priori’ knowledge of field experts can be embedded into the possibility space by the appropriate selection of feature classes (T_i), features, and compatibility relations ($\check{S}$). Accordingly, the method can utilize this ‘a priori’-given heuristic knowledge, while the continuous reevaluation of this knowledge base by the newly suggested and evaluated maximal combinations can also result in the supervision of originally applied equivalence classes, properties, and binary relations.

The functioning of RPS-based algorithm can be summarized as follows:

- At the beginning, based on experts’ knowledge, an initial RPS structure (e.g., the equivalence classes, the features within these classes, and the binary compatibility relations between the features) is determined.
- In the most sophisticated general case, the uncertain knowledge associated with the minimal (binary) relations can be described by the evaluation distribution of the formerly tested complete combinations (scenarios), containing the given two features. In the actually used simplified solution, the uncertain knowledge is described by the domain of this distribution, represented by a sub-interval $[m, M]$ within the $[0, 1]$ interval.

- At the beginning, all binary relations are evaluated with 'lack of knowledge', as a special description of "unknown case ($m = 1$ and $M = 0$).
- While learning from experts, these intervals are characterized by the values of the formerly tested worst (m) and best (M) evaluations of the given binary relations. An increasing number of evaluations results in widening intervals; sometimes, these sub-intervals may even cover the whole $[0, 1]$ interval. Nevertheless, the narrower sub-intervals contain usable uncertain knowledge about the applicability of the given binary relations.
- The selection of the subsequent features from the equivalence classes to the next complete combinations is algorithmically controlled by various selection strategies, as follows:
 - maxmax: maximize the upper bound M , i.e., the best value;
 - maxmin: maximize the lower bound m , i.e., the worst value;
 - maxave: maximize the average value;
 - unknown: prefer the still unknown binary relations (where $m = 1$ and $M = 0$);
 - minunc: minimize the uncertainty, i.e., prefer the shorter sub-intervals;
 - mintry: prefer the less frequently used elements;
 - random: prefer random choice from a subset of the above options.
- Against accumulating uncertainty, special reversible, irreversible, or drastic forgetting algorithms might be applied (in the actual work, it is not yet applied).
- In multi-objective (or multi-expert) evaluation, the Reevaluable Possibility Space can be multiplied for each objective (or expert). In this case, the intersections of the sub-intervals $[m, M]$ for the various objectives (or experts) express a consensus evaluation, while the conflict of objectives can also be derived algorithmically by subtracting the consensus from the aggregation. Next, the parallel evaluations follow based on the consensus-containing knowledge base, stepwise (in the actual work, it is not yet applied, only in the simplified toy example in Chapter 5.1).
- To decrease the experts' workload, consensus evaluation of multiple experts can be replaced by aggregation of their evaluations in the same knowledge base as we applied it in our work. However, it is worth mentioning that aggregation increases uncertainty.

As outlined in the objectives, the proposed methodology does not aim to identify a single optimal problem-solving method for a specific manufacturing task. Instead, combining

conventional and AI/ML-based methods and identifying tendentially better problem-solving approaches under uncertainty are the goals. While the RPS supports the comparison and partial ordering of complete combinations, the ranking must be viewed in context and is not absolute, given the task-specific nature of solving manufacturing problems. To suggest new combinations from the lattice of equivalence classes is an essential feature of the proposed methodology. However, by design, it cannot suggest the combination of two or more properties from the same equivalence class. If that turned out to be required, then it might be solved by introducing double or multiple combinations from the same equivalence class as new properties in the same class.

5. RESULTS

5.1 Simplified illustrative example for possibilistic learning from the evaluated lattice of equivalence-classified properties

Before the implementation and analysis of RPS for proposing and assessing computational tools in the manufacturing industry, we show a simplified illustrative example about the essential features and main steps of possibilistic learning from the evaluated lattice of equivalence-classified properties.

The equivalence classification and evaluation feedback-based learning serves as a tool for computer-assisted discovery of possible new combinations, associated with an inherent possibilistic uncertainty. Starting from the triviality that suggested new combinations cannot be considered as a random phenomenon, their uncertain feasibility cannot be characterized by probabilistic measures. While reasoning under probabilistic uncertainty requires large, representative samples to make reliable statistical inferences about the given real world random events, this work deals with possibilistic uncertainty (as we explained in the context of the method selection), that does not rely on sample representativeness in the frequentist sense, but instead models the possibilistic uncertainty in sense of selection elements from a set, according to continuous logic of $[0, 1]$ values (Dubois et al., 2000). To place this in context, a brief explanation is as follows.

Possibilistic uncertainty was defined by the pioneering work of Lotfi Zadeh (Zadeh, 1978). Based on this, probabilistic uncertainty quantifies uncertainty using probability distributions over outcomes, while possibilistic uncertainty represents uncertainty according to fuzzy logic. In the classic didactic example of Zadeh, he contrasts probability distributions with possibility distributions, through the question „How many eggs did Joe use to eat for breakfast?“ In this term, probability requires a precise allocation of probability mass over all values (1, 2, ..., 10 eggs), with a mode of 2-3. Another question is: „How many eggs is Joe able to eat for breakfast?“ This allows statements like 1, 2,...10 eggs with a possibility value of 1, next it starts to go down converging to 0 (impossible), say beyond 20.

In the following simplified and fictitious ("toy") case study, the Reevaluable Possibility Space based learning algorithm:

- is built from arbitrarily chosen property classes, properties and binary relations between the classified properties;

- focuses on the effect of multiple expert evaluations; and
- excludes the author's preconception.

The steps of implementation and execution are as follows:

1. Definition of an arbitrary number of optional property equivalence classes, say p1, p2 and p3.
2. Definition of arbitrarily chosen compatibility relations amongst these property classes, say:

[p1,p2]

[p1,p3]

[p2,p3]

Considering the actually chosen 3 equivalence classes, all possible relations are allowed, however, in other examples, some relations cannot be applied, only the connectedness of each class has to be guaranteed.

3. Definition of an arbitrary number of optional properties in these property classes, say
three properties of p1_1, p1_2, p1_3 in equivalence class p1;
four properties of p2_1, p2_2, p2_3, p2_4 in equivalence class p2;
two properties p3_1, p3_2 in equivalence class p1.
4. Definition of binary relations between the properties of compatible property classes according to #2, evaluated by 'lack of knowledge', initially.

The binary relations are described by the `rmin()` facts. Their definition and actual example-related binary relations are described in Appendix A3.

These initial exemplary binary relations are prepared for a single evaluator, called e1. In case of multiple evaluators, each of them has a dedicated binary knowledge base, initiated by making copies from the above set of relations, and renaming them as e1, e2, e3, etc. These individual knowledge bases can be used for automatic generation of the hypothetical knowledge base of their consent (see later on, more detailed).

Considering the 3 equivalence classes, in this example, all possible binary relations are allowed, however, in other cases, some binary relations between some properties can be omitted, only the connectedness of all properties has to be guaranteed.

The 'lack of knowledge' is represented by declaring $V_{min} = 1$ and $V_{max} = 0$ evaluations in sense of the reevaluable lattices and of the underlining four-valued logic for the marginal case of discrete $\{0,1\}$ evaluation of complete combinations.

5. The already existing and newly generated maximal combinations, synthesized from the #4 binary relations, are represented by the $rmax()$ facts. Their definition and actual list of $rmax()$. facts can be found in Appendix A3.

To demonstrate the absence of bias in the evaluations, in this “toy” example we generated and saved with randomly selected hypothetical "true" evaluations for all possible complete combinations, in advance.

Having initiated the learning algorithm, the knowledge base-evaluating steps start with one (actually the first) maximal combination.

Furthermore, to demonstrate the absence of the effect of any bias in the evaluations, in this fictitious example, we simulated the possibly erroneous and/or subjective evaluation of experts by randomly selected values (from a configurable expert-specific interval) around the hypothetical "true" evaluation of the given complete combinations. For example, we defined the random evaluations of $e1$ experts according to the following rules:

$$\begin{aligned} \text{expert}(e1, E0, E) \text{ if} \\ L \text{ is } E0 - E0/3, \\ U \text{ is } E0 + (1 - E0)/3, \\ \text{random}(L, U, E). \end{aligned} \tag{22}$$

where

- $E0$ = the previously generated hypothetical "true" value of the given combination;
- L = the lower bound of the random range for the evaluators around the $E0$ value;
- U = the upper bound of the random range for the evaluators around the $E0$ value;
- E = the simulated, possibly erroneous and subjective evaluation of the experts.

This means that simulated expert $e1$ is assumed to evaluate the given combination with a partly subjectivity originated (actually random) error within the (configurable) $[L, U]$ interval, where L represents one third of the $[0, E0]$ interval and U represent one third of the $[E0, 1]$ interval. The analogously calculated, but different errors for the applied 8-experts case are declared in Appendix A3.

6. Starting from the above-described principles and datasets, the cyclically repeated basic execution steps of the RPS algorithm (for each expert) are the following:

- (i) reevaluation of the $\text{rmin}()$ binary relations by the expert evaluation of the non-evaluated, newly suggested $\text{rmax}()$ combinations);
- (ii) proposition of new (or optionally repeated $\text{rmax}()$ combination(s) from the respective binary knowledge base.

The reevaluation of a given binary relation is illustrated in Figure 4. The initial ‘lack of knowledge’ changes for a given real value, when this knowledge element appears first in a complete combination. Later on, this binary relation may participate in multiple combinations of different values, resulting in an increasing interval, expressing also the increasing possibilistic uncertainty.

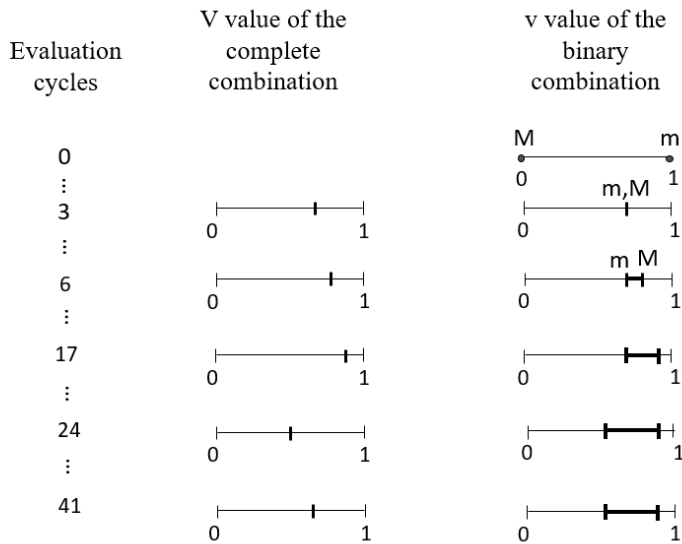


Figure 4 Illustration of the changing evaluation of a given binary relation

The selection of a binary relation from a given class is based on ranking of them according to various strategies (as described in Chapter 4.2.3). Figure 5 shows an example for ranking of binary relations within a given class, according to the strategy maxmin (where m is expected to be maximum).

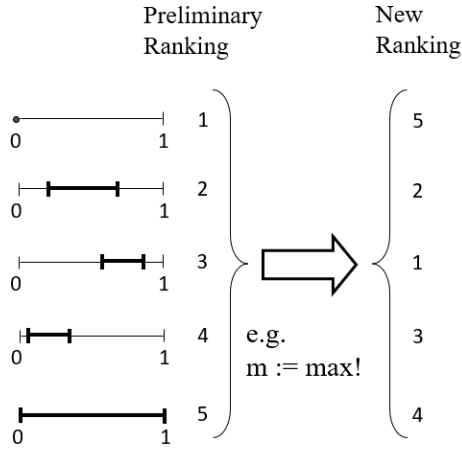


Figure 5 Illustration of ranking of binary relations within a given class, according to the strategy *maxmin*

Our simplified example study involves 60 steps of reevaluations, followed by propositions of new complete combinations by multiple (3 or 8) simulated experts, having individual reevaluated knowledge bases.

In the following case studies, we applied random order of selected pairs of property classes and randomly applied selection strategies from the set of *maxmax*, *maxave*, *maxmin* and *unknown*.

Figure 6 summarizes the various kinds of uncertainties of knowledge for evaluator *e1*, in case of 3 evaluators. The uncertainties of unknown and known binary relations, as well as the total uncertainty, summarizing their absolute values are calculated according to Equations (18) – (21). Figure illustrates that the uncertainty of ‘lack of knowledge’ decreases, and the uncertainty of knowledge increases, while the total uncertainty first decreases, and after a minimal value, increases toward a limit value.

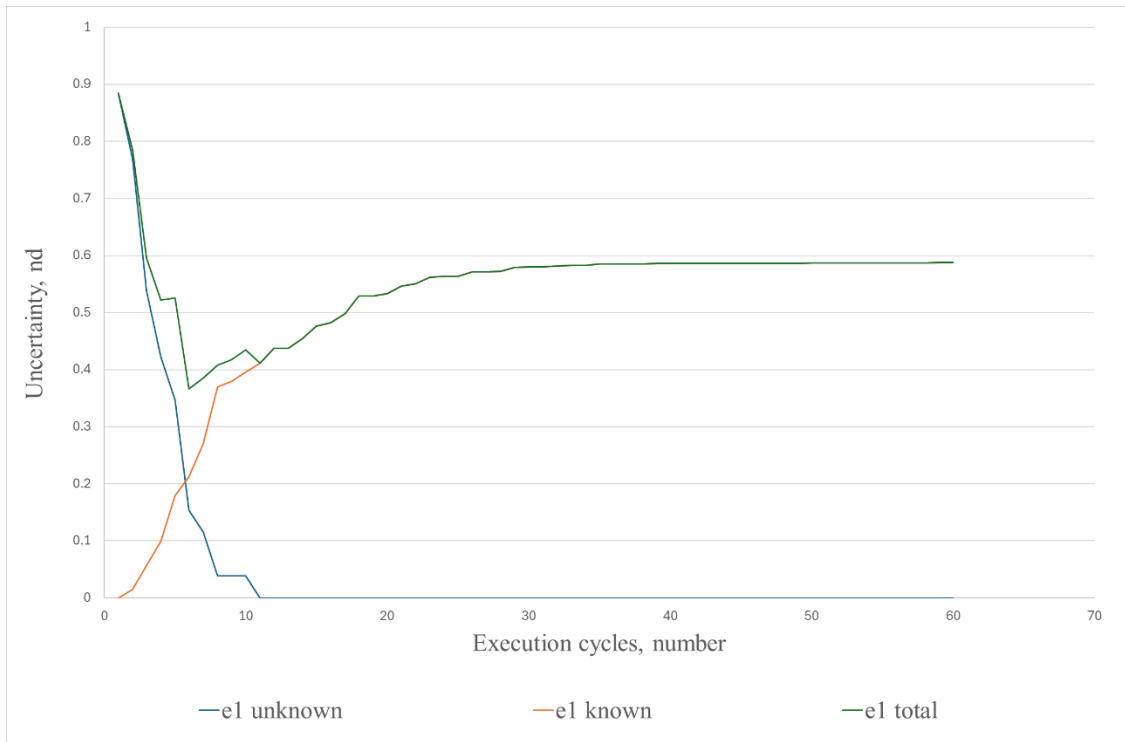


Figure 6 Summarized uncertainties of unknown and known binary relations, as well as the total uncertainty for evaluator *e1*, in case of 3 evaluators

This expresses that the uncertainty from the ‘lack of knowledge’ turns into uncertainty of accumulating knowledge. Meanwhile, it results in a continuous accumulation of useful information in the respective *rmin* knowledge base. However, an inherent part of uncertainty comes from the fact that, even in the case of sound evaluation of complete combinations, the value of lower-order (finally binary) relations is increasingly uncertain, because they participated in various complete combinations of different values. Additional uncertainty originates from subjective and erroneous evaluations of experts.

This fictitious case study makes it possible to study only the inherent uncertainty coming from the fact that the uncertainty of binary relations is inherent, because they participated in various complete combinations of different values. This can be calculated by applying an “absolutely knowledgeable” fictitious expert, who finds out the “true” value of maximal combinations (i.e. $E = E_0$ in (Eq. 22)). Accordingly, this run characterizes only the above outlined inherent uncertainty, without the uncertainty related to the erroneous and subjective evaluation of expert(s). Figure 7 compares this inherent uncertainty with the uncertainty of knowledge calculated in previous case study made for simulated expert *e1* in case of 3 experts. The difference characterizes the subjective, erroneous uncertainty of *e1*.

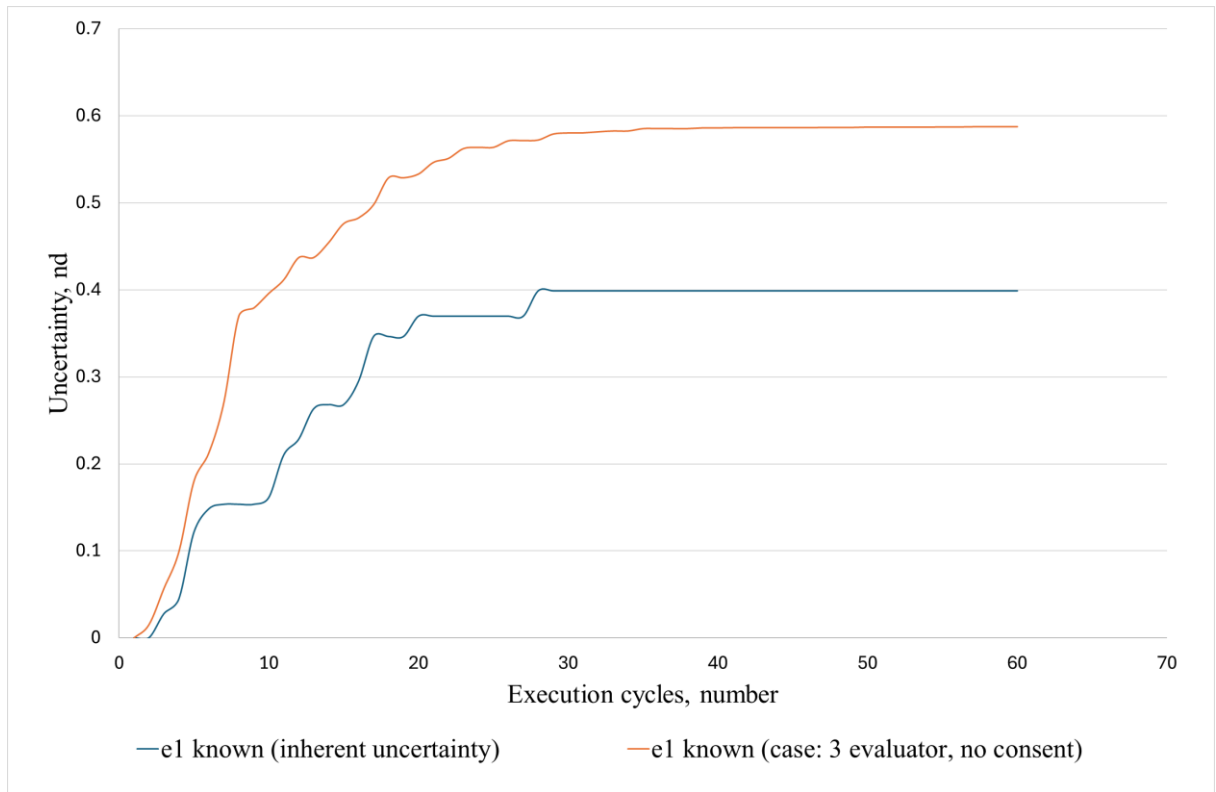


Figure 7 Comparison of inherent uncertainty with the uncertainty of knowledge calculated in previous case study, made for simulated expert *e1*, in case of 3 experts with no consent

As it can be seen in Figure 7, two-third of possibilistic uncertainty come inherently from the situation that the minimal elements participated in various complete combinations of different values.

7. Next, we analyze the effects of number and quality (error, subjectivity) of involved simulated experts.

Figure 8 shows the uncertainty of available knowledge for 3 evaluators. Each case go toward a limit value, while their difference is the consequence of differently “erroneous” and “subjective” simulated evaluations.

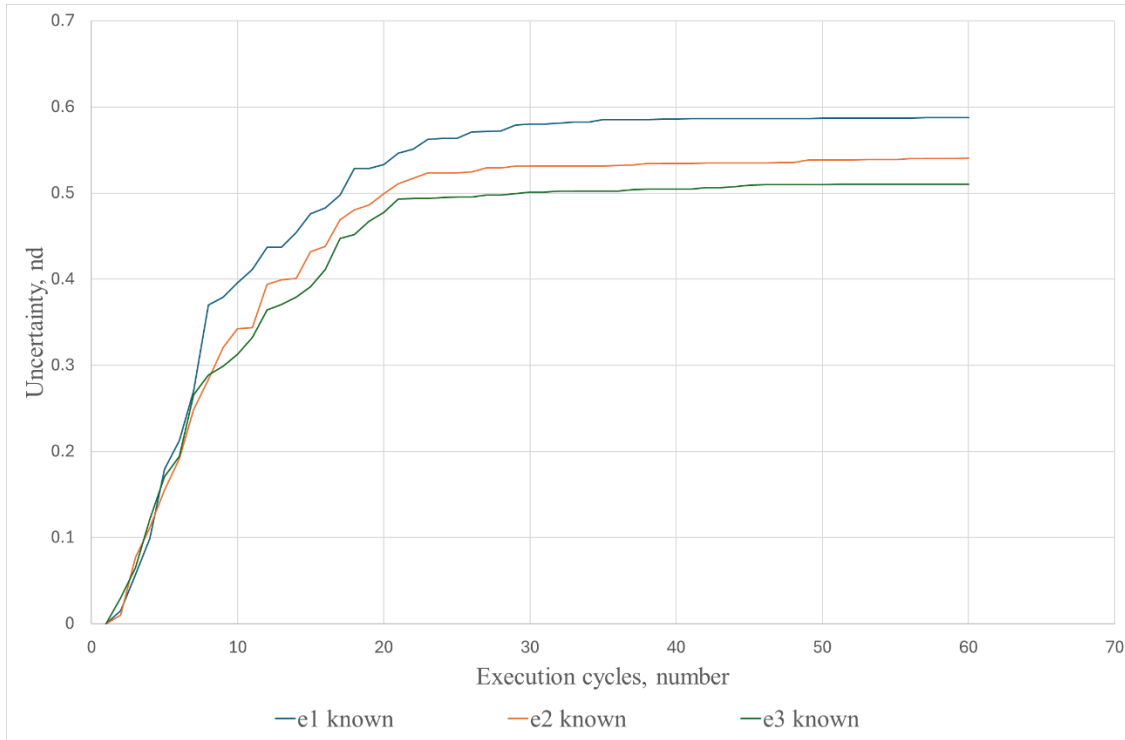


Figure 8 Uncertainty of available knowledge for 3 evaluators

The simulations with 8 evaluators do not show any major differences. According to Figure 9, more evaluators represent an increasing range of uncertainty in the development of their individual binary knowledge bases.

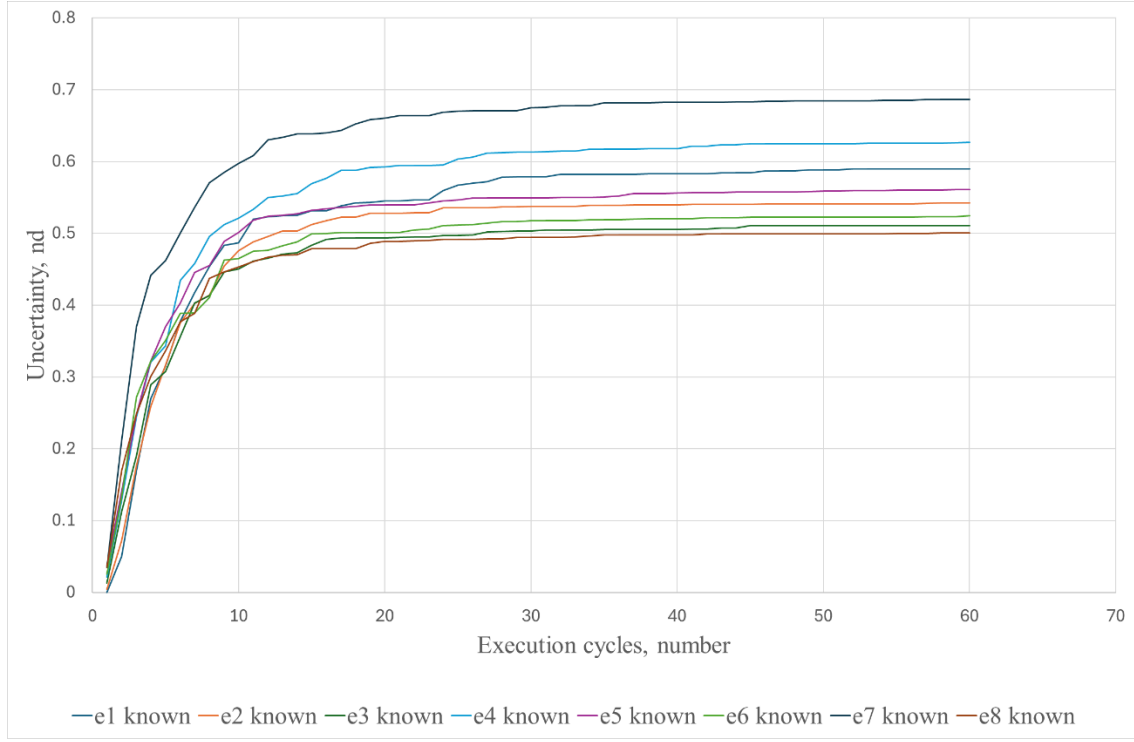


Figure 9 Uncertainty of available knowledge for 8 evaluators

8. The consent-embedded execution of the RPS algorithm involved arbitrarily chosen 10×6 steps of reevaluations by multiple (3 or 8) simulated experts, followed by proposing of new complete combinations from each expert's knowledge base. Every 10 steps were followed by automatic generation of the consent-representing knowledge base, next the knowledge base of each expert were replaced for the consensus-expressing knowledge base.

In the interpretation of consensus, we calculate for each binary relation the MaxMin and the MinMax values, as follows:

$$\text{MaxMin} = \max(\text{Min}(e1), \text{Min}(e2), \dots \text{Min}(eN))$$

$$\text{MinMax} = \min(\text{Max}(e1), \text{Max}(e2), \dots \text{Max}(eN))$$

Convex consensus exists if and only if

$$\text{MaxMin} \leq \text{MinMax}$$

If this condition is fulfilled, then the $\text{Min}(eX)$ and $\text{Max}(eX)$ values of the given binary knowledge bases are replaced for MaxMin and MinMax, respectively.

Figure 10 summarizes the various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators, utilizing stepwise consent. As can be seen, consensus decreases uncertainty, so the limit value is also less than in the no-consensus case (Figure 6).



Figure 10 Various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators, utilizing stepwise consent

Figure 11 illustrates the effect of stepwise consensus in case of 3 evaluators.

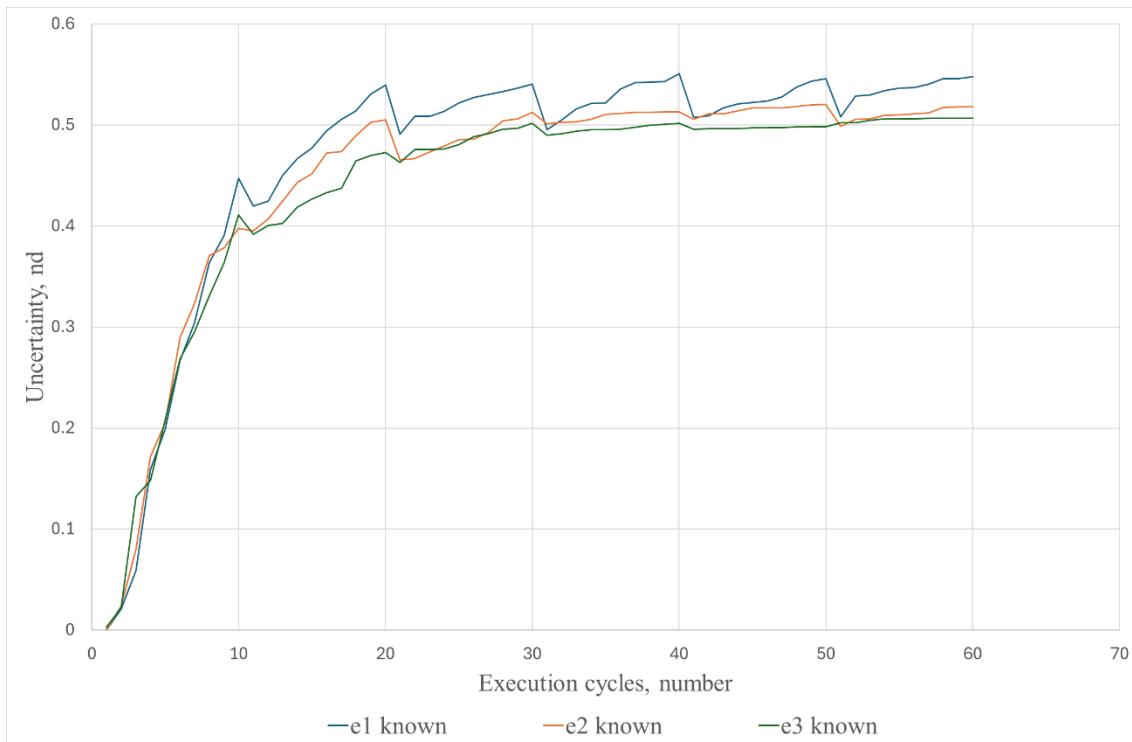


Figure 11 Effect of stepwise consensus, in case of 3 evaluators

Figure 12 illustrates the effect of stepwise consensus, in case of 8 evaluators about uncertainties of already evaluated, known relations.

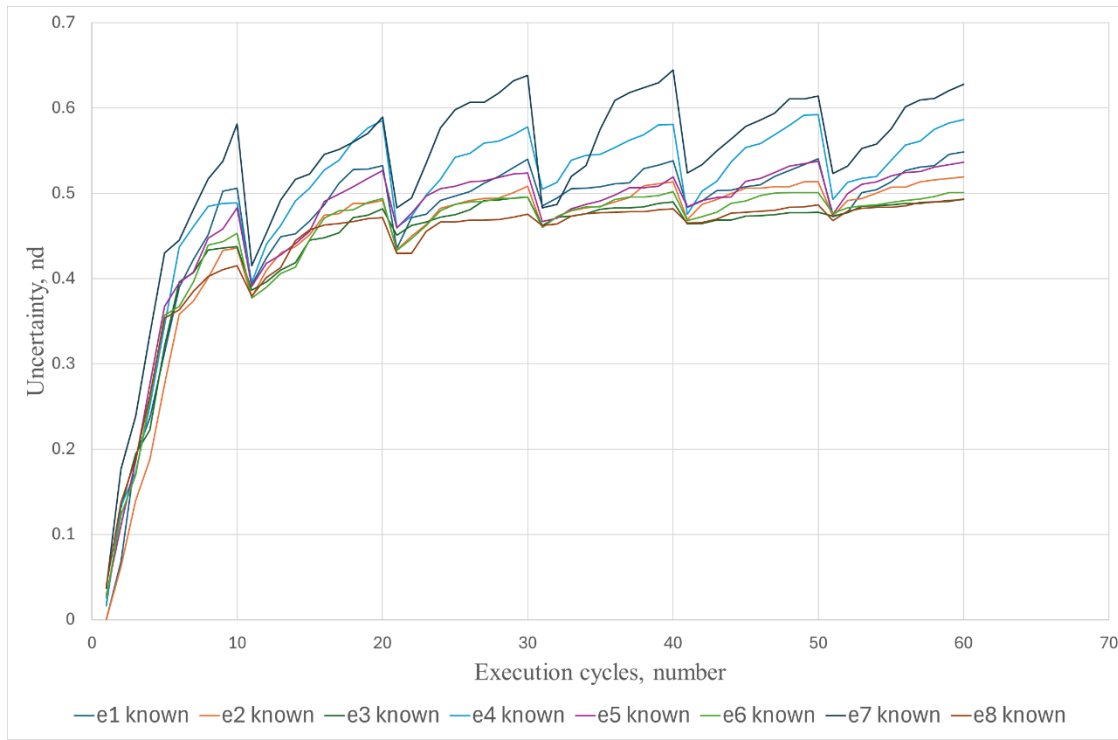


Figure 12 Effect of stepwise consensus for various types of uncertainties, in case of 8 evaluators

The lessons from this simplified fictitious “toy” example can be summarized as follows:

- The reevaluated lattice of equivalence classified properties works also in the case of simulated randomized evaluation of arbitrarily chosen property classes and properties. The essential functioning is the consequence of the structure behind the evaluation feedback, as the structure of binary relations carries foundational ‘a priori’ knowledge.
- The applied method resulted in fast elimination of ‘lack of knowledge’, and in a similar increasing tendency of uncertain knowledge for all studied cases.
- The randomized feature of the example illustrates that the essential functioning of reevaluable possibility space is not affected by the number of evaluating experts.
- The increasing uncertainty of already evaluated known binary relations consists of two parts. The major part comes from the inherent uncertainty, expressing the situation that a single binary relation can participate in multiple complete combinations with different values. The second part expresses the possible erroneous and/or subjective evaluations of the experts. This can be higher as the number of experts increases, as it incorporates their diverse experiences, while aggregation of evaluations will be affected by the more uncertain expert opinions.
- The cyclically generated consensus, followed by continuation of evaluation feedback of each expert, starting from the same consent-expressing evaluations, can cyclically decrease the uncertainty.

- The example illustrates the main steps of the implementation of RPS, as well as emphasizes the possibilistic character of handling uncertainty in knowledge discovery.

More details about the implementation and results of this simplified, illustrative example can be found in Appendix A3.

5.2 Equivalence-classified properties of the investigated set of computational problem-solving methods for sustainable manufacturing

Equivalence classes provide a mutually exclusive classification of the properties used to describe situations in manufacturing that require problem-solving, on the one hand, and computational problem-solving methods, on the other. Each equivalence class contains alternative properties, each representing different aspects of a task-method combination. Each possible task-method combination must consist of one and only one property from each equivalence class to form a complete combination. Since this is a fundamental requirement of the framework, the definition of the equivalence classes and their properties requires careful and knowledgeable consideration. This approach is heuristic; nonetheless, it represents a systematic way to formalize the initial a priori knowledge used to structure the possibility space (Weber & Varga, 2025a).

5.2.1 Property classes and properties of manufacturing tasks

The manufacturing tasks to be solved can be described by the property classes and properties presented in Figure 13:

P1 – Task of application	P2 – Level of application	P3 – Phases of problem solving	P4 – Development goal
• p1_1 Product design • p1_2 Process design • p1_3 Process control • p1_4 Fault diagnosis and troubleshooting • p1_5 Operation planning • p1_6 Operation scheduling • p1_7 Supply chain management • p1_8 Maintenance • p1_9 Quality control • p1_10 Worker training	• p2_1 Process unit • p2_2 Technology (a connected set of process units) • p2_3 Company • p2_4 Industry • p2_5 Trans-sector	• p3_1 Preparatory analysis of possibilities • p3_2 Implementation and testing of method/tool • p3_3 Execution and evaluation of method/tool	• p4_1 Safety • p4_2 Economic objective • p4_3 Environmental impact • p4_4 Circularity

Figure 13 Property classes and properties of manufacturing tasks

As described in Figure 13, the manufacturing tasks are structured into four equivalence

classes, which are the application tasks, the levels of application tasks, the stages of problem-solving tasks, and the evaluation objectives of the assigned task. This develops a structural methodology enabling a clear and aggregate understanding of manufacturing processes and associated management aims.

Property class P1, Task of application, includes all manufacturing tasks based on their functional role in the manufacturing process. Such an organization, by functional task, helps with resource allocation and work streamlining. It ensures that each task of the manufacturing process is handled by specialized teams or systems designed towards reaching optimal efficiency and thus maximize performance.

Established productivity benefits have been documented in the specialization in manufacturing tasks organization (Pisano & Shih, 2009). Not switching teams also has a higher chance of detecting problems in quality earlier, and corrective cycles are reduced. In AI/ML-supported environments this is particularly impactful: retraining or reconfiguring a model is expensive, and switching of tasks that interrupt a stable run may propagate latency effects over a production line. A task-based manufacturing organization should be implemented with careful thought in the distribution of resources and selection of technologies (Brynjolfsson & McAfee, 2014; Pisano & Shih, 2009). It is worth mentioning that the referenced sources do not fully agree on how specialization affects technology adoption. Pisano & Shih are skeptical of offshoring specialized capabilities, while Brynjolfsson & McAfee are more optimistic about substituting with technology.

The property class 2, level of application, organizes manufacturing tasks according to the scale of their operation, from the fineness of the individual units of the process up to the wide range of influence of trans-sectors. Such differentiation enables customized approaches based on the context of specific challenges and opportunities at various levels that would result in both optimal and suitably scaled solutions.

The unit level is the most basic level in manufacturing, as it involves the performance of particular tasks, namely mixing, heating, or assembling. The aim of this level is to achieve efficiency, safety, and quality in the individual operations. The unit should ensure operational excellence, reduce downtime, and be linked with upstream and downstream units to achieve the best performance.

The technology level is organized by combining units of processing to achieve a greater manufacturing purpose, e.g., a production line. The focus of this level is to maintain a smooth integration and coordination of process units. The challenges at this level are to strike the right

balance between flexibility and efficiency, be able to adjust to new production technologies, and technology lifecycles (Groover, 2020).

The company level describes all the manufacturing of one organization. It aligns the manufacturing plan with the overall business goals, including market responsiveness, sustainability, and innovation. Key considerations are resource allocation, talent management, and corporate culture. Market forces, the provision of a well-trained workforce, and innovation and operational excellence all serve to challenge this level (Huggins & Izushi, 2011).

At the industry level, the unit of analysis is the sector and not the company. Limitations that no one company has absolute control over are competitive dynamics, common infrastructure, and regulatory regimes. Issues of sustainability and global competition meet here, where industry-wide innovation, standardization, and cooperation are major concerns.

The trans-sector level relates to how manufacturing connects to other sectors of the economy and society, such as cross-industry relationships and community and environmental effects (Elkington, 1997). At this level, some of the focus areas are sustainability and the circular economy. One of the challenges that receives little attention in the literature is the issue of stakeholder alignment. The coordination between companies is challenging internally; collaboration across sectors entails entities with altogether different incentive systems.

A flat, function-only framework would not capture the way decisions at the process unit level carry implications to company and industry levels, and the reverse is true, too. The classification by level of application allows analyzing manufacturing tasks at the appropriate scale. The cases in Chapter 5.2 illustrate this concretely.

Property class 3, ‘Phases of problem solving,’ clusters tasks into chronological stages of manufacturing.

At the very beginning of problem-solving, before committing resources to implementation, the preparatory analysis of possibilities aims to explore and analyze potential solutions. This phase involves identifying and describing the problem, researching existing solutions, brainstorming new possibilities, and assessing the feasibility of various identified options.

The method/tool implementation/testing aims to convert the selected solutions into feasible applications within the manufacturing process. The step involves designing or modifying the tools or processes, deploying the tested solution into the operating environment, and examining its effect. It is a step of observing the implementation, gathering performance information, and comparing the results with goals (Schilling, 2022).

As a matter of fact, the definitions between these phases are seldom crisp. In many instances analyzed in this dissertation, there was overlap between preparatory analysis and implementation. It was particularly widespread in explanatory AI/ML-based applications, where testing itself contributed to the analytical value of narrowing down the initial statement of the problem. Instead of viewing that as a failure of the classification, it is better to think of it as a feature of the implemented AI/ML-based problem-solving that the framework must support. Several conditions needed to negotiate these overlaps include disciplined project management, effective communication, and a willingness to revise in midcourse (Kerzner, 2017).

Property Class 4, ‘Evaluating Objectives,’ organizes manufacturing tasks according to primary objectives. This classification acknowledges the multifaceted nature of manufacturing priorities beyond mere economic outcomes and includes safety, environmental sustainability, and circularity. The importance of these properties is backed by the results of the systematic literature review described in Chapter 3.

The health and safety of employees, customers, and communities is the focus of the safety goal. This involves introducing effective safety measures, threat assessment, and emergency readiness, and balancing cost and efficiency.

The economic target focuses on enhancing financial performance by reducing costs, increasing revenue, and improving efficiency in resources, such as by improving productivity, minimizing waste, and taking advantage of economies of scale. This categorization has placed quality control within the specific economic objective rather than classifying it as a distinct category. The rationale is simple: in AI/ML-led manufacturing, data integrity and model performance determine quality results. Computational performance and economic performance cannot be divided. This has some practical implications for the analysis of the cases in Chapter 5.3. The main issues in this level are how to work through economic lows and highs, dealing with supply chain complexity, and maintaining financial investment in innovation without risking short-term financial stability.

Environmental impact is a multidimensional goal. The primary goal is to reduce the environmental footprint of the manufacturing processes. This includes cutting down emissions, resource conservation, and sustainability practices. Along with it, an understanding of the handprint of the product you produce or the service you offer is essential in the decision-making of the downstream partners to decrease their footprint. Optimization of environmental impact is achieved by integrating environmental sustainability into the manufacturing processes and

the harmonization of environmentally motivated endeavors with economic agendas (Elkington, 1997).

Circularity incorporates the values of the circular economy, with a zero-waste lifecycle where products are designed to be recycled and reused, as well as recovered. Circularity is an advanced form of sustainability, focusing on the efficiency of resources in processes and product development. It aims to have a longer product life cycle and regenerate natural systems. Its obstacles are market demand for circular products and implementation of the principles of circularity in the supply chain (Geissdoerfer et al., 2017).

The evaluating objectives classification is not only organizational. It demonstrates the strains that manufacturing practitioners are expected to manage. Circularity and economic performance do not go hand in hand. A number of examples in Chapter 5.3 discuss circular design decisions that may have upfront cost premiums that only become economically justifiable through analysis at the technology or industry level rather than at the process-unit level. To cope with them, it is necessary to incorporate cross-functional cooperation and the desire to perceive sustainability and circularity as parameters of design, not compliance (Krajnc & Glavič, 2005).

5.2.2 Property classes and properties of computational problem-solving methods

The problem-solving methods can be characterized by the property classes and properties, presented in Figure 14.

P5 – Conventional problem-solving methods	P6 – AI/ML-based problem-solving methods
• p5_1 (Dynamic) simulation • p5_2 Model-based Predictor-Corrector (MPC) • p5_3 Stochastic Process Control (SPC) • p5_4 Mixed Integer Linear Programming (MILP) • p5_5 Mixed Integer Non-Linear Programming (MINLP) • p5_6 Finite Element Analysis (FEA) • p5_7 Data Mining and Analytics • p5_8 Computer-aided Design (CAD) • p5_9 Computer-aided Manufacturing (CAM) • p5_10 Real-time Monitoring and Control Systems • p5_11 Simulation-based Optimization • p5_12 Expert Systems • p5_13 N/A	• p6_1 Convolutional Neural Networks (CNNs) • p6_2 Support Vector Machines (SVMs) • p6_3 Random Forests • p6_4 k-Nearest Neighbors (k-NN) • p6_5 Fuzzy Logic and Reasoning • p6_6 Genetic Algorithms (GAs) • p6_7 Evolutionary Learning Strategies • p6_8 N/A

Figure 14 Property classes and properties of computational problem-solving methods

The computational problem-solving methods relevant to the study fall into two clusters: conventional and AI/ML-based methods. These are not simply sequential stages in the evolution of the field. In the manufacturing context examined, they operate in parallel, each addressing problem types that the other handles less well. Understanding where their capabilities overlap and diverge is necessary for evaluating the complete combinations.

The class 5 problem-solving methods, namely, Conventional problem-solving methods, incorporate algorithm-based approaches, which are based on deterministic and rule-based principles (Horvath & Rudas, 2004; Oduguwa et al., 2005). The main tools under this category are optimization algorithms, scheduling models, and simulation models. They are strong in traceability: with a clear problem with constant parameters, they generate good, consistent results. The linear programming of supply chain management is an illustrative case wherein the material sourcing and routing medium can be combined in terms of cost and emissions optimization. The weakness lies in the fact that these approaches do not work as effectively in a situation where there are changes in the parameters of the problem in the course of operation, as may occur in dynamic manufacturing settings.

The sixth equivalence class is called AI/ML-based problem-solving methods and refers to data-based approaches, where the distinguishing feature is the ability to respond to emerging data instead of acting in accordance with a set of rules (Yin & Kaynak, 2015; Zheng et al., 2019). This flexibility is advantageous in manufacturing settings where conditions of operational change can be recalibrated more quickly than in deterministic models. The applications reflected in this paper include predictive maintenance, anomaly detection, and real-time optimization. The fitting of the problem, rather than the ability, is the relevant differentiation in conventional methods. AI/ML-based procedures have associated costs: they need reasonable amounts of training data, their results are less interpretable, and when deployed, conditions differ significantly from those during training. These restrictions were not accidental in the combinations reviewed: they determined which types of tasks were to be treated with which computational methods.

The two kinds of computational problem-solving methods are infinitely useful to sustainable manufacturing and provide complementary solutions. Although traditional approaches offer a strong and stable framework for optimizing efficiency, AI/ML-based approaches introduce the flexibility and vision that is required to navigate the complex world of productivity and sustainability. The combination enables manufacturers not only to produce operational excellence but also to move towards a more sustainable and environmentally friendly future.

5.2.3 Interpretation of the property classes in the context of manufacturing systems

The property classes in Chapters 5.2.1 and 5.2.2 provide a structured representation of tasks, organizational levels, problem-solving phases, objectives, and computational problem-solving methods relevant to industrial manufacturing. Although the classification is presented in an abstract form to support the compatibility lattice structure introduced in Chapter 4, each property class corresponds directly to well-established concepts in manufacturing engineering and digital production systems.

The first property class, Task of application (P1), represents the types of decision problems encountered in manufacturing. The tasks listed in Figure 4 cover the major functional domains of industrial production systems. These domains correspond to the core activities discussed in manufacturing engineering literature, where decision-support methods are applied to improve product development, production efficiency, quality assurance, and supply chain coordination (Groover, 2020; Segreto & Teti, 2014). For example, process control tasks involve monitoring and regulating production parameters, while maintenance tasks may involve predictive diagnostics and condition monitoring of equipment.

The second property class, Level of application (P2), reflects the hierarchical organization of manufacturing systems. Industrial production systems are typically structured in multiple layers, ranging from individual process units and production lines to company-wide operations and industry-level ecosystems. This hierarchical view is widely used in production engineering and industrial systems analysis (Groover, 2020). Figure 4 illustrates the level differentiation that enables the computational problem-solving methods to be attributed to the organizational scope in which they are employed. An example is that process monitoring algorithms can be run on a machine or process-unit level, whereas the optimization of the supply chain or analysis of sustainability can involve company-wide or cross-industry decisions.

The third property class, Phases of problem solving (P3), describes the stage within the lifecycle of applying a computational method. The phases reflect a typical engineering workflow in which computational tools are first assessed for feasibility, then implemented and validated, and finally applied in operational decision-making. Industrial settings have similar stages of engineering design and project implementation practices (Kerzner, 2017). By distinguishing these phases, the framework captures the fact that computational methods may be used not only during operational decision-making but also during earlier design and evaluation stages.

The fourth property class, Development goal (P4), represents the objectives that motivate the application of computational methods. The selected goals reflect the multidimensional objectives of sustainable manufacturing systems. Contemporary industrial systems are shifting towards a balance between economic efficacy and environmental and societal commitments, in line with the concepts of sustainability, including both the triple-bottom-line concept (Elkington, 1997) and integrated sustainability assessment models (Krajnc & Glavič, 2005). Including those objectives in the classification ensures that the generated combinations of computational problem-solving methods can be evaluated with respect to both economic and sustainability-related goals.

The fifth and sixth property classes represent the computational problem-solving methods themselves, separated into conventional methods (P5) and AI/ML-based methods (P6). This difference is associated with the theoretical disjunction made in Chapter 4.1, in which conventional approaches can be termed model-based, rule-based, or analytically defined, whereas AI/ML-based approaches can be termed data-driven learning.

Taken together, the six property classes describe the essential dimensions of computational problem-solving in manufacturing: what problem is addressed, where in the organization it occurs, at which phase of the problem-solving process it is applied, which objectives are pursued, and which computational methods are used. By combining these dimensions, the framework can represent a wide variety of practical manufacturing scenarios. For example, a complete combination may represent the application of a random forest model for predictive maintenance at the process-unit level during the execution phase of problem solving with the objective of improving operational reliability and economic performance.

The aim of introducing the classification in abstract form is not to isolate the framework of the practical manufacturing applications, but to allow the representation of the combinatorial relations among tasks, goals, levels of organization, and methods for solving the problems systematically. This structured representation forms the basis for the compatibility lattice and the RPS introduced in Chapter 4, which allows the systematic evaluation and generation of feasible methodological combinations.

5.3 Initial compatibility structure of the binary relations between the property classes

First, the existing relations between the property classes must be determined. By default, it may be a full network, as can be seen in Figure 15. Nevertheless, some relations between two equivalence classes may not contain additional restrictions, because the other binary relations

determine the whole “design space.” This can be refined based on the literature analysis and the heuristic knowledge of the experts.

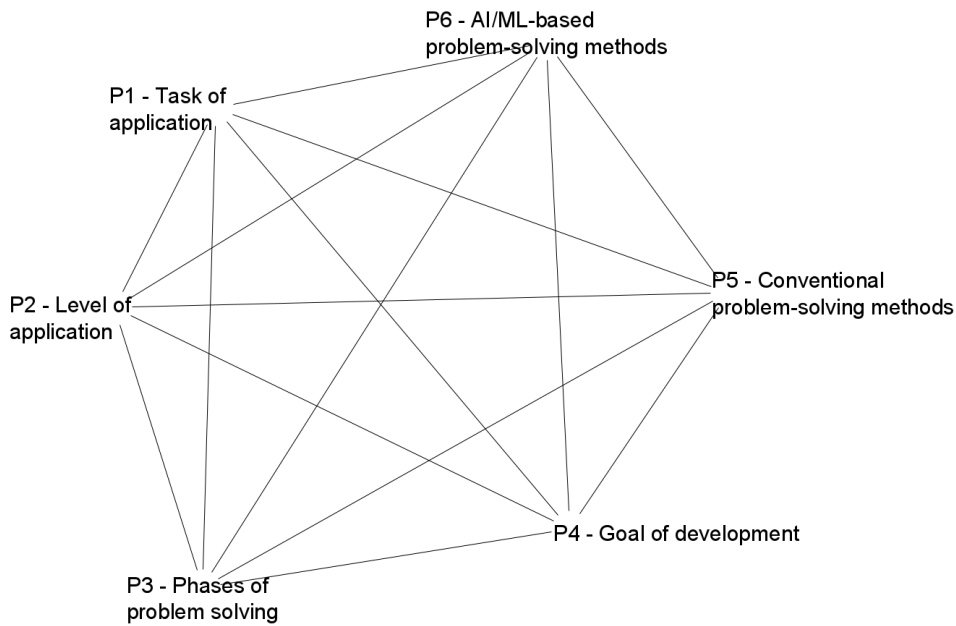


Figure 15 Relations among properties of P1 – P6 equivalence classes

Figure 15 shows the binary relations among properties of the equivalence classes P1 through P6. The six nodes are connected by all possible 15 edges to begin with. While irrelevant binary combinations between property classes are possible, they would manifest in irrelevant binary combinations of properties.

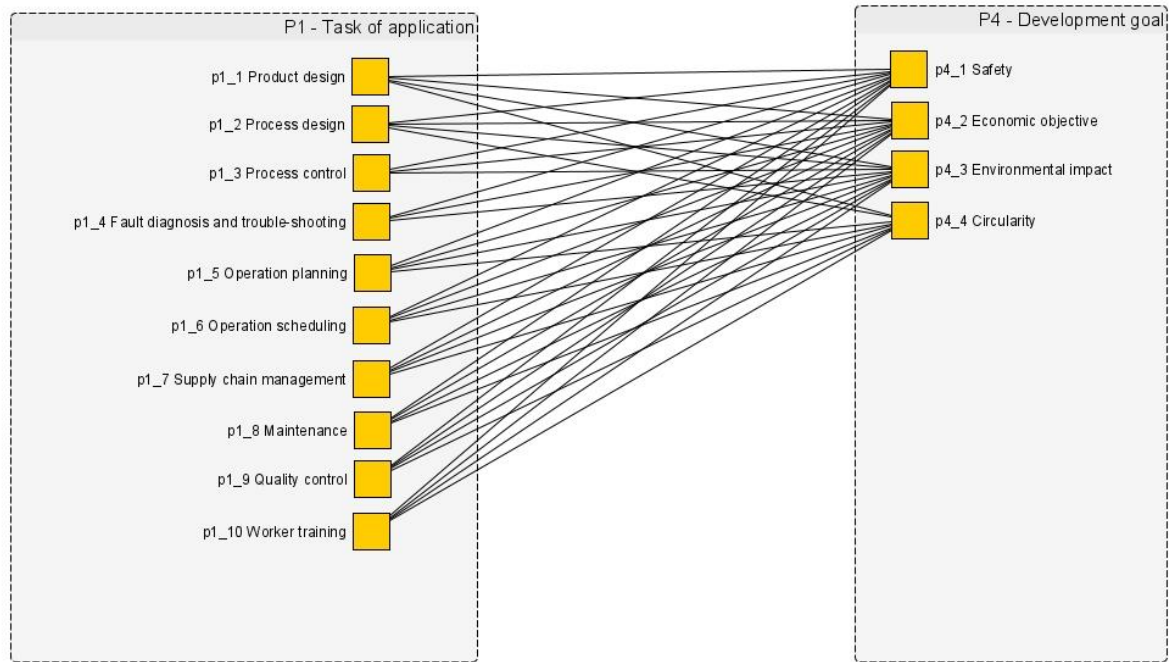


Figure 16 Exemplary graphic representation of the binary combinations of properties of the property classes P1 and P4

Exemplary for all possible binary relations, Figure 16 shows binary relations among properties of the property classes P1 and P4. At this level of granularity, irrelevant combinations have been removed, i.e., the relations between process control (p1_3) and circularity (p4_4).

Initially, the binary combinations can be evaluated by the “unknown” value in sense of the four-valued logic. This means the apparently paradox [1,0] case, where $m = 1$ and $M = 0$. Accordingly, the given binary combination may be evaluated by 1 also in the worst case and by 0 also in the best case. Consequently, any evaluation of a newly studied combination can initiate the first realistic evaluation for the given binary combination. An experienced expert can modify the zero knowledge by adding other initial evaluations in form of heuristically declared $[m,M]$ intervals, of course. Nevertheless, automatic learning from the evaluated complete combinations may be more convenient.

5.4 Description of the complete combinations

According to the applied methodology, the complete combinations, describing an actual task and the applied conventional and/or AI/ML-based methods, are described by sets of properties, containing one and only one element from each property class. In line with the SWI-Prolog representation of applied algorithms, the complete combinations are described by `rmax()` predicates, according to the following syntax:

`rmax(XY, ID, Nu, Plist, Evallist)`

where

XY = identifier of expert evaluator;

ID = identifier integer;

Nu = number of occurrences;

Plist = list of properties, participating in the given combination;

Evallist = contains evaluation of the given complete combinations.

As an example, an actual `rmax()` predicate is shown as follows:

`rmax(dw, 11, 1, [p1_10, p2_3, p3_3, p4_2, p5_11, p6_6], [0.7])`

In this complete combination example, the expert with the initials ‘dw’ evaluated the 11th complete combination of a set of complete combinations that had not been evaluated previously.

Its description is as follows:

- Task of application: worker training (p1_10);
- Level of application: company (p2_3);
- Phase of problem solving: execution and evaluation of method/tool (p3_3);
- Evaluation objective: economic objective (p4_2);
- Conventional problem-solving method: Simulation-based Optimization (p5_11);
- AI/ML-based problem-solving method: Genetic Algorithms (p6_6).

Listing one property of each property class, this complete combination describes an application of executing and evaluating a tool for a case of worker training at the company level. An economic objective is pursued, and the combined problem-solving methods to be evaluated are Simulation-based Optimization and Genetic Algorithms.

In this example, the expert with the initials ‘dw’ evaluates the given complete combination with a 0.7 on a scale from 0 to 1.0.

Referring to the industrial park example from page 11, these properties combined describe the challenge accurately to look for feasible and useful methods or method combinations:

- p1_5 Operation planning
- p2_2 Technology (connected set of process units)
- p3_1 Preparatory analysis of possibilities
- p4_2 Economic objectives

5.5 Description of the binary relations between the classified properties

Binary relations represent the relevance and the uncertain evaluation of the actual pair of properties, belonging to two equivalence classes. Again, in line with the SWI-Prolog

representation of applied algorithms, the binary relations are described by `rmin()` predicates according to the following syntax:

`rmin(XY, ID, YN, Freq, Classes, Properties, Vmin, Vmax, Unc, Vav, Forget)`

where

`XY` = identifier of Expert evaluator;

`ID` = identifier integer;

`YN` = to be considered by default `y` (it means that it is considered during the evaluation);

`Freq` = number of reevaluations (by default 0 at the very beginning, and will be overwritten during the evaluations);

`Vmin` = minimal value (initial ‘lack of knowledge’ considered by 1);

`Vmax` = maximal value (initial ‘lack of knowledge’ considered by 0);

`Unc` = uncertainty, `Unc = Vmax-Vmin`;

`Vav` = average value, `Vav = sign (Unc) * average (Vmax, Vmin)`;

`Forgetting` = the status of the optional forgetting steps, by default 0.

As an example, an actual `rmin()` predicate is shown as follows:

`rmin(dw, 456, y, 13, [p2,p6], [p2_4, p6_2], 0.1, 0.8, 0.7, 0.45, 0)`

In this example, the expert with the initials ‘dw’ evaluated the relation #456, which has been evaluated 13 times. The property classes P2 Level of application and P6 AI/ML-based problem-solving methods are included, specifically the binary relation between the two properties `p2_4` Industry and `p6_2` Support Vector Machines. For this binary relation, the pessimistic value `Vmin` is 0.1 and the optimistic value `Vmax` is 0.8; the uncertainty `Vmax-Vmin` is therefore 0.7. The average value is 0.45.

At the beginning, all binary relations are considered to be ‘unknown.’ It corresponds to the apparently paradoxical evaluation of $m = 1$ and $M = 0$, while in this case the given binary combination is evaluated by 1 also in the worst case and by 0 also in the best case. Accordingly, if the first evaluation for a binary relation appears, then the ‘unknown’ case will change for the case of $m = M = \text{first value}$. The same binary relation may occur in other combinations of various values later on, so this will outline a $[m, M]$ interval for the evaluation of the given binary relation. Furthermore, in the initial ‘unknown’ case, `Freq` is 0, `Unc` is -1 , and `Vav` is -0.5 . A complete initial knowledge base, containing 733 binary relations between the 15 property

classes, can be found in the file of Rmin_0.pl in the respective repository (Weber & Varga, 2025b).

5.6 Overview of the algorithm in the sense of Reevaluable Possibility Space

The simplified scheme of the evolving knowledge base (Weber & Varga, 2025a) (in form of reevaluated binary relations) can be seen in Figure 17.

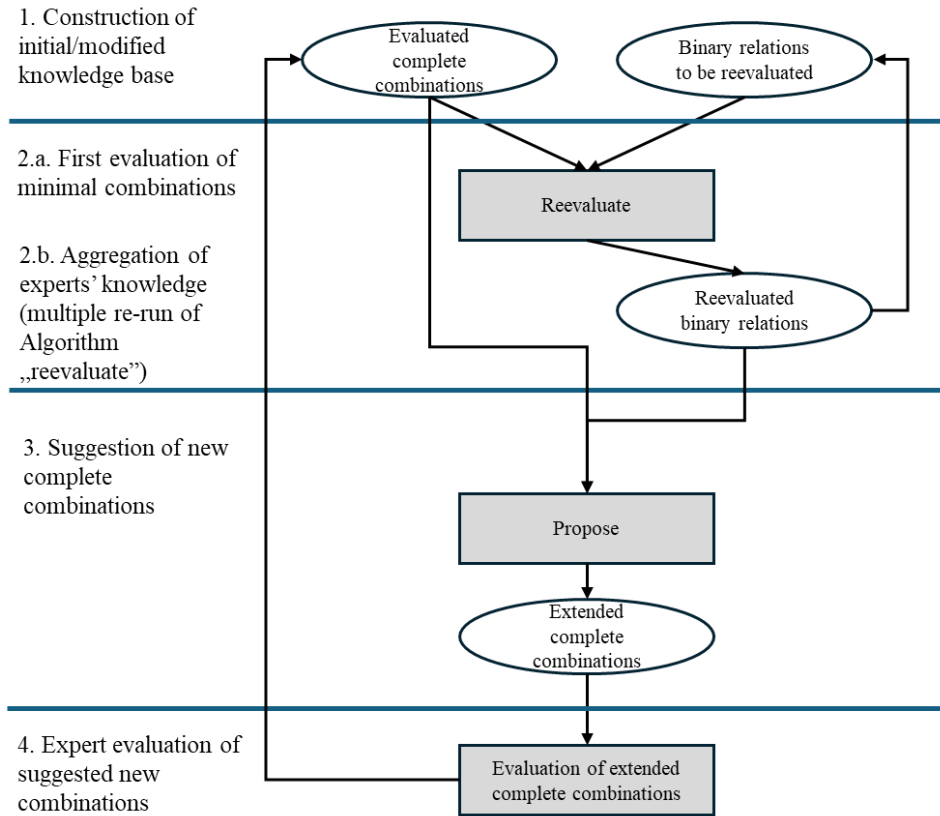


Figure 17 Simplified scheme of the evolving knowledge base of evaluated binary relations.

Ellipses symbolize Rmax files, containing an actual set of possible alternative solutions (scenarios) in the form of rmax() complete combinations, as well as Rmin files, containing rmin() binary relations, corresponding to the actual state of the evolving knowledge base. Rectangles symbolize computational algorithms or a human expert's activity. More detailed interpretations of these elements are the following:

Evaluated complete combination: First input file of Algorithm 'reevaluate' that contains previously evaluated rmax() combinations only. At the very beginning it was filled by 300 complete combinations, evaluated preliminarily. Also, it serves as an input file of Algorithm 'propose'.

Binary relations (knowledge base) to be reevaluated: Second input file of Algorithm 'reevaluate' that contains an actual state of rmin() relations. At the very beginning it was filled

with ‘lack of knowledge’ (with ‘unknown’ $m = 1$ and $M = 0$) evaluations according to file `Rmin_0.pl`.

Reevaluate: Reevaluates the above binary relations by the above evaluated combinations, resulting in a reevaluated knowledge base. Essentially, the algorithm takes all the possible property pairs in each complete combination and modifies the respective m and M values, as well as the other attributes in the knowledge base.

Reevaluated binary relations (knowledge base): The result of Algorithm ‘reevaluate’ (that incorporates the upgraded knowledge) and the first input file of Algorithm ‘propose’.

Propose: Extends the available complete combinations with new ones in the knowledge of the reevaluated binary relations of the last knowledge base. The configuration and parameterization of the algorithm are discussed in Chapter 4.2.

Extended complete combinations: The result of Algorithm ‘propose’ and the input for expert’s evaluation.

Evaluation of suggested new combinations: Made by (optionally multiple) experts, while the new combinations will obtain a normalized value, heuristically.

The evaluated complete combinations can be used to reevaluate the knowledge base (see the arrow, representing the respective feedback on the left-hand side of Figure 17. The last version of reevaluated binary relations can be applied for further reevaluation of the knowledge base (see the respective feedback on the right-hand side of Figure 17.

In this work, three experts were involved in a single workflow of the above procedure, and their specific knowledge was aggregated (resulting in higher uncertainty, of course). Involving a larger number of experts would improve the robustness of the knowledge base, but at the same time it may increase uncertainty due to the variability in expert judgements. This can be avoided by the application of consensus and conflict calculations, as it is illustrated in the toy example (Chapter 5.1) and suggested for inclusion in future work.

5.7 Computational representation of the algorithm

The above discussed algorithm was implemented in SWI-Prolog language (<https://www.swi-prolog.org/>). The program code has been developed as a special application of the theoretical framework of RPS, described in Chapter 4.2. The developed source code is presented in Supplementary Material S2. The source code and the executable program, together with the applied data files can be found in the respective repository (Weber & Varga, 2025b).

The actual run of the program is controlled by a defined task() predicate, according to the following syntax:

task(Program, From1, From2, To, Input_parameters)

where

Program = the algorithm to be executed;

From1, From2 = the names of input files in the necessary syntax (i.e. within ‘’);

To = the name of output file in the necessary syntax (i.e. within ‘’);

Log = the name of the respective log file in the necessary syntax (i.e. within ‘’);

Input_parameters = the list of algorithm-dependent input parameters in the necessary syntax (i.e. within []).

Essentially, we use parameters only for the ‘propose’ algorithm.

The file containing the actually used task() predicates can be found in the respective repository ((Weber & Varga, 2025b) in Task.pl).

Two kinds of tasks are used, as follows:

- Algorithm = ‘reevaluate’, which modifies the knowledge base of minimal binary relations by the evaluations, coming from an actual set of evaluated complete combinations; and
- Algorithm = ‘propose’, which suggests new complete combinations, determined by the applied techniques:
 - Selection strategies of binary relation classes (random order or normal order), and
 - Selection strategies from the individual classes (random, maxmin, maxmax, maxave, unknown, minunc, mintry).

Program ‘reevaluate’ results in an updated knowledge base of binary relations, as well as some automatically calculated features, as follows:

- The number of binary relations;
- The number of unknown binary relations;
- The number of reevaluated binary relations;
- The absolute value of uncertainty for the unknown binary relations;
- The uncertainty of reevaluated binary relations.

Program ‘propose’ suggests a prescribed number of new complete combinations to be evaluated, as well as some automatically calculated features, as follows:

- The number of maximal relations;
- The number of suggested new maximal relations;

- The sum of maximal relations with repetition;
- The sum of suggested new maximal relations without repetition.

Running ‘propose’ depends on the respective [Actual_rmax_number, Suggested_new_rmax_number, Selected_ordering, Selected_strategy] parameters representing the following modes of the run:

- Actual_rmax_number: the number of the already existing evaluated combinations;
- Suggested_new_rmax_number: number of new combinations to be proposed;
- Selected_ordering of pairs of property classes: actual order or random order;
- Selected_strategy: maxmax or maxmin or maxave or unknown or minunc or mintry or random. Essentially, random is limited to maxmax or maxmin or maxave or unknown.

Reproducibility of randomization can be solved by tracking the run in an automatically generated log file.

5.8 Workflow of case studies

The overall workflow of the case studies is summarized as follows. Corresponding files can be found in the respective database (Weber & Varga, 2025b) to ensure reproducibility of the work.

5.8.1 Construction of an initial empty knowledge base and the initial set of evaluated complete combinations

The workflow started from zero initial knowledge about the evaluation of complete combinations and minimal binary elements. Accordingly, all binary relations were characterized by the ‘lack of knowledge’ ($m = 1$, $M = 0$). After having finalized the structure of the knowledge base and having evaluated a preliminary set of complete combinations, we started from 300 complete combinations (Rmax_full1.pl in the repository of (Weber & Varga, 2025b), evaluated by DW, initially). Furthermore, we selected a subset of 50 (good and bad performing) solutions; additionally, involved experts were asked to evaluate this set of initial complete combinations blindly (so the experts did not know of each other’s evaluation results) with a real number from the $[0, 1]$ interval, as well as the new complete combinations later on, generated by the algorithm. Experts were informed about the criteria of the evaluation to ensure reasonable comparability. They were tasked to combine the following three aspects into a single scoring:

- Feasibility: Experts should consider whether the complete combinations can realistically be implemented, both technically and operationally.
- Possibility of success: Experts should consider the likelihood whether the specific complete combination will solve the problem, if implemented.

- Value added: Experts should consider the expected contribution or the complete combination toward the defined development goal.

In this way, this value had to express their general opinion about the given combination, i.e., the solution of a given class with a given conventional or AI/ML-based method (i.e., $V = 1$ if the expert agrees that it is a well-applicable solution, and $V = 0$ if expert considers solution inapplicable).

Involved experts included the following:

- Thomas Hohnloser, Chief Information Officer, Infraserv Wiesbaden GmbH & Co. KG. InfraServ Wiesbaden is the operating company of an industrial park in Wiesbaden Germany. <https://www.infraserv-wi.de/en/home/industrial-park.html>
- Steffen Robus, IT project leader, <https://www.deka.de/deka-group>
<https://www.linkedin.com/in/steffen-robust-b94011163/> (accessed on 4 December, 2025). Before the current occupation, Steffen Robus was General Manager, S-Servicepartner Consulting GmbH, the inhouse IT consultancy of the German Sparkassen.
- Dennis Weber, PhD candidate, author of this dissertation, and Sustainability Communications Manager at Corning Incorporated. Corning Incorporated is a global manufacturing company in the material science space.

The motivation for including the author as an expert is pragmatic and exploratory in nature: the author's 13 years of professional experience in manufacturing companies and the need to ensure internal consistency in the initial construction of the knowledge base. The author did not presume any expected result. He represented the knowledge accumulated from the literature review. The assumptions regarding the effectiveness of AI/ML methods are not from the author's subjective belief, but from the literature. The author did not have any preconceived hypotheses, just analyzed the given problem. Because the evaluated combinations are both numerous and complex, expert assessments may inevitably incorporate subjectivity. Nonetheless, these evaluations remain highly valuable, as they synthesize domain-specific knowledge and enable informed judgments across a wide array of multifaceted configurations. Although the model relies strongly on these assessments, the consideration of multiple experts' opinions can decrease the effects of subjectivity.

Involvement started with online one-to-one meetings, focusing on the presentation of the task, the introduction of the defined property classes, and the properties within them, as well as the utilized method for evaluation. Experts were involved based on a convenience sampling,

considering particularly their expertise, proven by many years of practical experience. Experts agreed to share their identity regarding this work with their written consent.

Accordingly, the algorithm started to run with the following information, structured in the respective files, as follows:

- Rmin_0.pl: The set of minimal combinations, containing 733 binary relations between the properties of the 15 property classes. Initially, it was filled with zero knowledge.
- Rmax_full1.pl, Rmax_DW1.pl, Rmax_TH1.pl, and Rmax_SR1.pl files (DW, TH, and SR refer to the initials of involved experts), which contain the 300 as well as the 50–50 maximal combinations, evaluated by the involved experts.

5.8.2 First reevaluation of minimal combinations

Based on the previously mentioned files, the ‘reevaluate’ task of the algorithm results in the reevaluated knowledge bases Rmin_full1.pl, Rmin_DW1.pl, Rmin_TH1.pl, and Rmin_SR1.pl. These files started to be filled with reevaluations of the minimal relations that have obtained calculated values, based on their performance in the maximal combinations.

5.8.3 Aggregation of experts’ knowledge

In these steps, the opinions of various experts are aggregated in the knowledge base, with execution of the respective ‘reevaluate’ tasks. During this procedure, the ‘reevaluation’ is repeated in three consecutive steps, using the following files:

- Rmax_TH1.pl and Rmin_DW1.pl: The 50 maximal combinations, evaluated by expert TH, reevaluate the knowledge base of minimal combinations, coming from the former step, based on expert DW’s evaluation. This run results in the Rmin_DW1TH1.pl.
- Next, Rmax_SR1.pl reevaluates the formerly obtained Rmin_DW1TH1.pl, resulting in the Rmin_DW1TH1SR1.pl file that considers and aggregates the opinion of the third expert, represented in the (possibly) widening intervals in the binary combinations in the new Rmin_DW1TH1SR1.pl.
- Finally, the Rmax_full1.pl complete combinations reevaluate the former Rmin_DW1TH1SR1.pl, considering all the uncertain knowledge of the intervals of the binary relations in Rmin_DW1TH1SR1full1.pl.

It is to be noted that the aggregation process is commutative, associative, and idempotent, so the result is invariant for the order of aggregating steps.

5.8.4 Proposition of new complete combinations

In the knowledge of a Rmin set of aggregated binary relations and a Rmax file of former set of rmax() maximal combinations, the ‘propose’ algorithm suggests new possible maximal combinations, resulting in an extended Rmax file to be evaluated by the experts.

5.8.5 Expert evaluation of suggested new combinations

The experts evaluated these new suggested combinations. In the actual case study, blind expert evaluation of the 50 complete combinations was conducted, so the experts did not know of each other’s evaluation results.

5.9 Evolution of the binary knowledge base

Table 7 summarizes the case studies. Accordingly, it contains the name of the actual Rmin knowledge base, the name of the respective Rmax file (from which Rmin was reevaluated lastly), the acronym of the evaluating expert, the new Rmax files, containing the newly proposed rmax() combinations, as well as the assessment of the actual state of the Rmin file. This assessment considers:

- the number of binary relations,
- the number of unknown binary relations (expressing the temporary ‘lack of knowledge’),
- the number of reevaluated binary relations,
- the absolute value of the uncertainty of unknown binary relations,
- the uncertainty of reevaluated binary relations, as well as
- the sum of uncertainty, which is the sum of uncertainty for the unknown and reevaluated binary relations.
- Uncertainty values were calculated according to Equations (18)-(21). Table 7 and Figure 9 illustrate the average uncertainty of ‘lack of knowledge’ (unknown binary relations), the average uncertainty of reevaluated binary relations, as well as the sum of two kinds of uncertainty. It is to be noted that negative uncertainty of ‘lack of knowledge’ was replaced by its absolute value for the proper representation.

1 Table 7 Quantitative report on the evolution of the binary knowledge base

Cases	Name of Rmin file	Respective Rmax (on the basis of which Rmin was reevaluated)	Evaluator	Newly proposed Rmax	Number of binary relations	Number of unknown binary relations	Number of reevaluated binary relations	Normalized uncertainty of unknown binary relations	Normalized uncertainty of reevaluated binary relations	Normalized sum of uncertainty
C0	Rmin_0	---	---	---	733	733	0	1.00	0	1.00
C1	Rmin_DW1	Rmax_DW1	DW	Rmax_DW2_300 Rmax_DW2_50	733	310	423	1.00	0.14	0.50
C2	Rmin_TH1	Rmax_TH1	TH	Rmax_TH2_300 Rmax_TH2_50	733	310	423	1.00	0.15	0.51
C3	Rmin_SR1	Rmax_SR1	SR	Rmax_SR2_300 Rmax_SR2_50	733	310	423	1.00	0.09	0.47
C4	Rmin_Full1	Rmax_Full1	DW	Rmax_full2_300 Rmax_full2_100	733	28	705	1.00	0.40	0.43
C5	Rmin_DW1TH1SR1	Aggregation 1: Rmax_TH1 + Rmin_DW1 > Rmin_DW1TH1 Aggregation 2: Rmax_SR1 + Rmin_DW1TH1 > Rmin_DW1TH1SR1	---	Rmax_DW1TH1_2_300 Rmax_DW1TH1_2_50	733	310	423	1.00	0.48	0.70
C6	Rmin_DW1TH1SR1full1	Aggregation 3: Rmax_full1 + Rmin_DW1TH1SR1 > Rmin_DW1TH1SR1full1	---	Rmax_DW1TH1SR1_2_300 0 Rmax_DW1TH1SR1_2_50	733	28	705	1.00	0.47	0.49
C7	Rmin_DW1TH1SR1_2	Rmax_DW1TH1SR1_2ev_50	DW	Rmax_DW1TH1SR1_3_300 0 Rmax_DW1TH1SR1_3_50	733	155	578	1.00	0.39	0.52

C8	Rmin_DW1T H1SR1full1_2	Rmax_DW1TH1SR1full1_2ev_ 100	DW	Rmax_DW1TH1SR1full1_ 2_100 Rmax_DW1TH1SR1full1_ 2_300 Rmax_DW1TH1SR1full1_ 3_100 Rmax_DW1TH1SR1full1_ 3_300	733	12	721	1.00	0.52	0.53
C9	Rmin_DW1T H1SR1full1_4	Rmax_DW1TH1SR1full1_2evT H_100	TH	Rmax_DW1TH1SR1full1_ 4_450	733	7	726	1.00	0.55	0.56
C10	Rmin_DW1T H1SR1full1_5	Rmax_DW1TH1SR1full1_500	DW	---	733	6	727	1.00	0.58	0.58
C11	Rmin_DW1T H1SR1full1_6	Rmax_DW1TH1SR1full1_600	DW	---	733	2	731	1.00	0.61	0.61
C12	Rmin_DW1T H1SR1full1_7	Rmax_DW1TH1SR1full1_650	DW	---	733	0	733	0	0.62	0.62

The uncertainties of the binary relations in the knowledge base are developing with each reevaluation cycle, as Table 7 shows. In line with the analysis of the simplified illustrative example in Chapter 5.1, the uncertainty from the ‘lack of knowledge’ is decreasing from 1 to 0, while the uncertainty of the reevaluated binary relations slowly increases from 0 to 0.62. Nevertheless, the sum of these two kinds of uncertainties first decreases, then slowly increases, as shown in Figure 18 below. It expresses that the uncertainty from the ‘lack of knowledge’ turns into uncertainty of accumulating knowledge. Meanwhile, it results in a continuous accumulation of useful information in the rmin knowledge base. To further clarify, the increase in uncertainty at later learning stages does not indicate overfitting or model deterioration, but reflects the transition from ‘lack of knowledge’ to richer but more nuanced knowledge. The inherent uncertainty comes from the fact that, even in the case of sound evaluation of complete combinations, the value of lower-order (finally binary) relations is increasingly uncertain, because they participated in various complete combinations of different values. Additionally, uncertainty originates from subjective experts' evaluations and their aggregation.

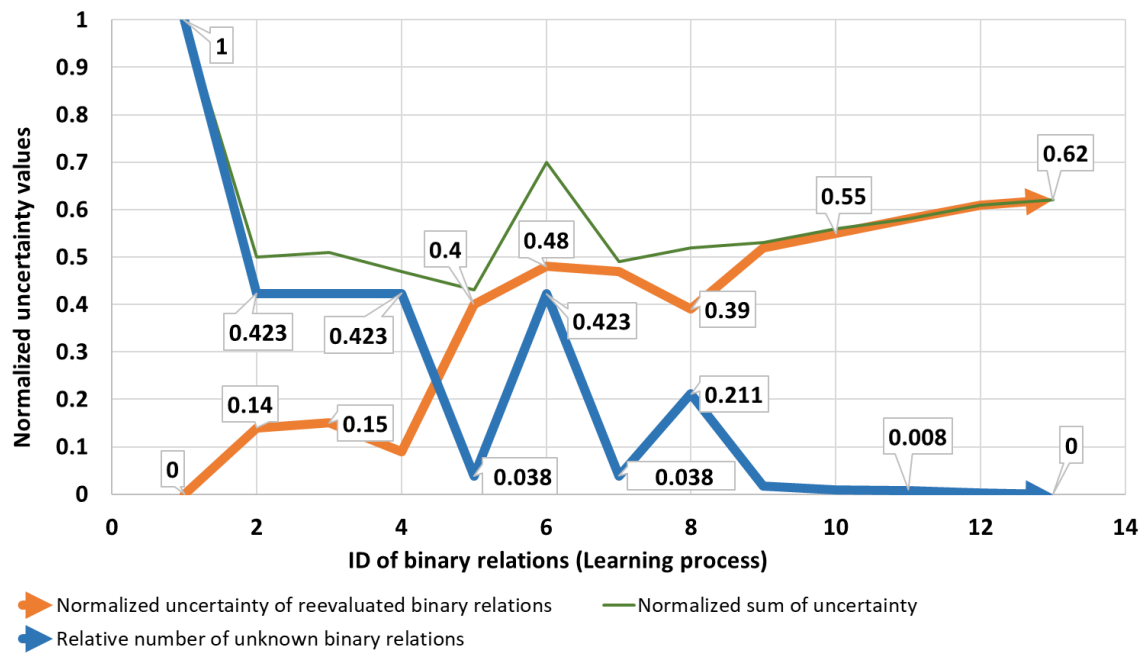


Figure 18 Development of uncertainty of binary relations

Comparing these results with the similar tendencies demonstrated in the fictitious “toy” example, the total uncertainty is a little bit higher (0.62, compared to the 0.54 value in the “toy” example in case of 3 evaluators). This comes partly from the aggregation of experts’ evaluations. Moreover, it comes from the larger possibility space, especially because the number of possible combinations are essentially higher, with respect to the number of binary relations. Anyway, the results outline the scalability of the applied methodology.

6. DISCUSSION

6.1 Evaluation and interpretation of the suggested complete combinations

With each round of evaluation, new complete combinations were suggested by the algorithm. Rarely repeating a suggestion, the algorithm grows its knowledge base.

In line with the accumulated knowledge in binary relations, the consecutive runs suggest continuously good and bad solutions, while the good solutions may contain advantageous combinations. Analyzing the 50 newly suggested and evaluated combinations of Rmax_DW1TH1SR1full1_650, 10 combinations (20%) are evaluated ≥ 0.8 , 20 combinations (40%) ≥ 0.7 . Those are high percentages of good combinations compared to the initially suggested combinations, underlining the conclusion.

Listed in Table 8 are the top-ten and the lowest-ten combinations of the 50 newly suggested combinations in Rmax_DW1TH1SR1full1_650, while these example combinations illustrate the actual state of the knowledge base to be developed further.

Table 8 The ten highest and ten lowest evaluated newly suggested combinations of Rmax_DW1TH1SR1full1_650

P1 Task of application	P2 Level of application	P3 Phases of problem solving	P4 Development goal	P5 Conventional problem-solving method	P6 AI/ML-based problem-solving methods	Expert evaluation
p1_6 Operation scheduling	p2_1 Process unit	p3_1 Preparatory analysis of possibilities	p4_4 Circularity	p5_1 (Dynamic) simulation	p6_8 N/A	0.9
p1_8 Maintenance	p2_1 Process unit	p3_3 Execution and evaluation of method/tool	p4_1 Safety	p5_12 Expert systems	p6_8 N/A	0.9
p1_1 Product design	p2_5 Trans-sector	p3_1 Preparatory analysis of possibilities	p4_1 Safety	p5_9 Computer-Aided Manufacturing (CAM)	p6_1 Convolutional Neural Networks (CNNs)	0.9
p1_4 Fault diagnosis and troubleshooting	p2_5 Trans-sector	p3_3 Execution and evaluation of method/tool	p4_1 Safety	p5_9 Computer-Aided Manufacturing (CAM)	p6_6 Genetic Algorithms (GA)	0.8
p1_10 Worker training	p2_3 Company	p3_1 Preparatory analysis of possibilities	p4_3 Environmental impact	p5_7 Data Mining and Analysis	p6_6 Genetic Algorithms (GA)	0.8

p1_6 Operation scheduling	p2_4 Industry	p3_1 Preparatory analysis of possibilities	p4_3 Environmenta l impact	p5_10 Real- time Monitoring and Control Systems	p6_3 Random Forests	0.8
p1_9 Quality control	p2_2 Technolog y (a set of connected process units)	p3_1 Preparatory analysis of possibilities	p4_2 Economic objective	p5_1 (Dynamic) simulation	p6_8 N/A	0.8
p1_3 Process control	p2_4 Industry	p3_1 Preparatory analysis of possibilities	p4_2 Economic objective	p5_7 Data Mining and Analysis	p6_8 N/A	0.8
p1_6 Operation scheduling	p2_3 Company	p3_2 Implementatio n and testing of method/tool	p4_1 Safety	p5_11 Simulation- based Optimization	p6_6 Genetic Algorithms (GA)	0.8
p1_4 Fault diagnosis and trouble- shooting	p2_2 Technolog y (a set of connected process units)	p3_2 Implementatio n and testing of method/tool	p4_1 Safety	p5_12 Expert systems	p6_5] Fuzzy Logic and Reasoning	0.8
p1_10 Worker training	p2_4 Industry	p3_1 Preparatory analysis of possibilities	p4_1 Safety	p5_8 Computer- aided Design (CAD)	p6_1 Convolutiona l Neural Networks	0.3
p1_4 Fault diagnosis and trouble- shooting	p2_3 Company	p3_1 Preparatory analysis of possibilities	p4_2 Economic objective	p5_6 Finite Element Analysis	p6_1 Convolutiona l Neural Networks	0.2
p1_4 Fault diagnosis and trouble- shooting	p2_1 Process unit	p3_1 Preparatory analysis of possibilities	p4_2 Economic objective	p5_5 Mixed Integer Linear Programming	p6_1 Convolutiona l Neural Networks	0.2
p1_5 Operation planning	p2_2 Technolog y (a set of connected process units)	p3_1 Preparatory analysis of possibilities	p4_4 Circularity	p5_5 Mixed Integer Linear Programming	p6_8 N/A	0.2
p1_1 Product design	p2_1 Process unit	p3_3 Execution and evaluation of method/tool	p4_1 Safety	p5_5 Mixed Integer Linear Programming	p6_1 Convolutiona l Neural Networks	0.1
p1_9 Quality control	p2_2 Technolog y (a set of connected	p3_3 Execution and evaluation of method/tool	p4_1 Safety	p5_6 Finite Element Analysis	p6_1 Convolutiona l Neural Networks	0.1

	process units)					
p1_6 Operation scheduling	p2_1 Process unit	p3_1 Preparatory analysis of possibilities	p4_1 Safety	p5_9 Real-time Monitoring and Control Systems	p6_5 Fuzzy Logic and Reasoning	0.1
p1_10 Worker training	p2_1 Process unit	p3_1 Preparatory analysis of possibilities	p4_3 Environmental impact	p5_2 Model-based Predictor-Corrector	p6_3 Random Forests	0.1
p1_9 Quality control	p2_1 Process unit	p3_1 Preparatory analysis of possibilities	p4_2 Economic objective	p5_2 Model-based Predictor-Corrector	p6_4 k-Nearest Neighbors (k-NN)	0.1
p1_4 Fault diagnosis and troubleshooting	p2_3 Company	p3_1 Preparatory analysis of possibilities	p4_2 Economic objective	p5_6 Finite Element Analysis	p6_1 Convolutional Neural Networks	0.2
p1_6 Operation scheduling	p2_2 Technology (a set of connected process units)	p3_3 Execution and evaluation of method/tool	p4_2 Economic objective	p5_13 N/A	p6_8 N/A	0.0

Disproportionate occurrence of properties indicates above-average relevance to the development of the knowledge base, as they highlight advantageous and disadvantageous relations. To identify such patterns and enable a structured comparison, two groups of the 50 newly suggested combinations of Rmax_DW1TH1SR1full1_650 are considered: the top-ten and the lowest-ten evaluated complete combinations.

Disproportionate occurrences among the top-ten evaluated complete combinations include:

- p1_6 Operation scheduling (4 out of ten combinations)
- p3_1 Preparatory analysis of possibilities (6 out of 10 combinations)
- p4_1 Safety (4 out of ten combinations)
- p6_8 N/A (4 out of ten combinations, which means in these cases no AI/ML methods are combined with conventional methods).

Disproportionate occurrences among the lowest-ten evaluated complete combinations include:

- p2_1 Process unit (5 out of ten combinations)
- p3_1 Preparatory analysis of possibilities (7 out of 10 combinations)
- p4_1 Safety (4 out of ten combinations)

- p6_1 Convolutional Neural Networks (5 out of ten combinations).

While these observations describe the distribution of properties among high- and low-evaluated combinations, a deeper understanding of the observed patterns requires interpretation in the context of method characteristics, application domains, and the structure of the RPS framework.

The results in Table 8 reveal distinct patterns in the highest- and lowest-evaluated suggested combinations. The following interpretation provides a differentiated explanation grounded in the characteristics of the applied computational problem-solving methods (Chapter 4.1), their typical domains of application in manufacturing (Chapter 3.1), and the structural properties of the RPS methodology (Chapter 4.2).

Interpretation of top-ranked combinations:

The salient feature of the highest-ranked combinations is the common lack of AI/ML-based approaches (p6_8 N/A), or, at most, hybrid configurations. Rather, conventional problem-solving methods (including dynamic simulation, expert systems, data mining, and computer-aided manufacturing) are common. This pattern can be explained by a number of complementary explanations.

First, most of the top-ranked combinations are related to the analysis stage (p3_1) preparation and system-level or cross-functional work. The problem contexts usually presuppose that they involve structured reasoning, transparency, and engineering knowledge integration. These requirements are well aligned with conventional approaches like simulation and expert systems because they are deterministic in nature and can be clearly interpreted (Giarratano & Gary D. Riley, 2004; Law, 2013). This is in line with the literature in Chapter 3.1, where explainable and model-based approaches are critical to the manufacturing decision-making at the early planning levels.

Second, traditional systems offer a clear depiction of the cause-and-effect relationships and system-level constraints, which are paramount in highly intricate manufacturing settings. As emphasized in Chapter 4.1, procedures like MPC, simulation-driven optimization, or CAD/CAM are used to work within clear physical and functional constraints. This capability is important in sustainable manufacturing contexts, where trade-offs between economic, environmental, and safety objectives must be explicitly managed (see Chapter 3.1.4).

Third, the observed pattern can be explained through the interaction with the RPS evolution mechanism. The RPS structure carries over judgments of whole combinations to judgments of relevance of binary relations in uncertainty (Chapter 4.2). Combinations that are constituted

based on well-known and generally used techniques have more uniform expert assessments, resulting in a smaller interval and more stable propagation of good evaluations. Concerts of the less established or more context-dependent methods can have a greater uncertainty, preventing their further promotion.

Fourth, hybrid combinations including AI/ML-based methods (e.g., CNNs or GAs), appear among the top-ranked solutions only when combined with conventional methods. This supports the interpretation, consistent with recent literature, that AI/ML-based methods are most effective when embedded within hybrid systems combining domain knowledge and data-driven learning (Von Rueden et al., 2023). In such configurations, AI complements rather than replaces conventional approaches.

Interpretation of low-ranked combinations:

The low-ranked combinations are characterized by a strong presence of AI/ML-based methods, particularly CNNs, as well as other data-driven approaches such as Random Forests and k-Nearest Neighbors. Here as well, several explanations can account for this observation.

First, CNNs and other AI/ML-based solutions are mainly used in high-dimensional data space tasks, including image or signal processing (Goodfellow et al., 2016). They have been successfully used in visual inspection of quality or detection of anomalies, which are well-documented. The low-ranked combinations place these methods, however, in the context of preparatory work, and in this case, the problem is not perceptual but conceptual and systemic. This discrepancy between the ability of the method and the requirement of the task offers a consistent reason behind their inferior evaluations.

Second, AI/ML-based methods demand big, high-quality data and consistent, stable data-generating mechanisms. Such data is usually either unavailable or ill-structured in early-stage problem-solving phases or in higher levels of organizations (e.g., industry or trans-sector level). This limitation is also discussed in Chapter 3.1, where data availability and quality are identified as key constraints for AI applications in manufacturing.

Third, AI/ML methods often exhibit lower interpretability than conventional methods, particularly in the case of deep learning models. In manufacturing environments, where safety, reliability, and regulatory compliance are critical (see also the section on OSHA and ISO standards in Chapter 3.1), limited explainability can reduce trust and thus influence expert evaluation negatively. This is consistent with findings on trustworthiness and transparency in AI systems (Thiebes et al., 2021).

Fourth, from the perspective of the RPS methodology, combinations involving AI/ML-based methods may be associated with higher uncertainty levels due to more heterogeneous expert evaluations. This leads to less stable propagation of positive evaluations and, consequently, lower evaluations in the generated suggestions.

Lastly, the occurrence of the very low-valued combinations, such as the ones where both conventional and AI/ML-based approaches are not available, proves the consistency of the evaluation framework. These combinations are incomplete or not plausible solutions in the given equivalence classes, and their low evaluation can be anticipated.

Insights from the comparison of top- and low-level-evaluated combinations:

Comparing the top and low-level-evaluated combinations reveals that performance is linked to the alignment between problem characteristics and method capabilities. The high-performing combinations are characterized by coherence between the type of task, stage of problem-solving, and the methodological approach, whereas low-performing combinations are characterized by the lack of coherence in these dimensions. These findings indicate that:

- Conventional methods dominate in structured, system-level, and early-phase tasks
- AI/ML-based methods are less suited for conceptual or preparatory problem-solving without sufficient data
- Hybrid approaches offer potential when method capabilities are complementary

These results can be compared to the literature presented in Chapter 3.4 because the current discovery methods can be classified as either probabilistic, optimization-driven, or evolutionary. In contrast to these methods, the RPS framework allows establishing such compatibility patterns explicitly by the evaluation of combinations, generated from the relations between the equivalence-classified properties.

Positioning of results and scope of validation:

The results that were presented do not represent the empirical validation of particular combinations of methods in manufacturing practice. The RPS methodology, as described in Chapter 4.2, is made to be a proposal-generating and learning method, facilitating decision-making by suggesting tentatively improved combinations, not delivering validated optimal solutions. The combinations suggested are not random. The consistency of the findings with the known facts about the capabilities of methods, data requirements, and application domains in manufacturing gives the results plausibility and interpretability. The RPS framework allows systematic searching of the space of methodological possibilities and helps to identify

promising directions. Individual combinations that have been empirically validated through simulation or industrial application would be a worthwhile avenue of future research, but are outside the scope of the current study.

6.2 Evaluation of the final state of the binary knowledge base

To evaluate the binary knowledge base, the 733 binary relations are ordered by the 15 class pairs, considering the Uncertainty, the Frequency, and the Average value of the respective binary relations (regarding the final state of the knowledge base, please see the Rmin/Rmin_DW1TH1SR1full1_7.pl in the github repository (Weber & Varga, 2025b)).

Analyzing the averages (average Frequency, average Uncertainty, and average Av. Value) for the paired classes, the average Av. Values appear to be trending around the middle value 0.5, independently from the Frequency and the Uncertainty. This, is in accordance with the last column of Table 9, underlines that in this phase of knowledge gathering, the elimination of ‘lack of knowledge’ introduces new relations participating in both good and bad combinations. In future applications, more iterative steps are recommended after having eliminated the ‘lack of knowledge’.

In the case of the average Frequency and the average Uncertainty, a strong correlation can be identified (correlation coefficient $r = 0.86$). Low average frequencies correlate with low uncertainty averages and vice versa, see Table 9 below.

Table 9 Correlation between average Frequency and average Uncertainty in paired classes.

Paired classes	Average Frequency	Average Uncertainty	Average Av. Value
P1 Task of Application - P5 Conventional problem-solving methods	23	0.41	0.51
P5 Conventional problem-solving methods - P6 AI/ML-based problem-solving methods	29	0.45	0.5
P1 Task of application - P6 AI/ML-based problem-solving methods	38	0.6	0.51
P2 Level of application - P5 Conventional problem-solving methods	47	0.59	0.51
P4 Development goal - P5 Conventional problem-solving methods	58	0.63	0.48

P1 Task of application - P2 Level of application	60	0.74	0.5
P3 Phases of problem solving - P5 Conventional problem-solving methods	76	0.70	0.48
P2 Level of application - P6 AI/ML-based problem-solving methods	76	0.72	0.47
P4 Development goal - P6 AI/ML-based problem-solving methods	95	0.77	0.47
P1 Task of application - P3 Phases of problem solving	101	0.80	0.5
P1 Task of application - P4 Development goal	102	0.81	0.5
P3 Phases of problem solving - P6 AI/ML-based problem-solving methods	127	0.79	0.47
P2 Level of application - P4 Development goal	152	0.87	0.5
P2 Level of application - P3 Phases of problem solving	203	0.89	0.5
P3 Phases of problem solving - P4 Development goal	254	0.93	0.5

The correlation indicates that an increased Frequency of binary relations among property classes leads to an increased Uncertainty. This is a consequence of aggregation instead of reaching consensus among the experts, explicitly. According to the experiences of the simplified case study in Chapter 5.1, consensus can be solved by the independent parallel knowledge base development by the experts, followed by an algorithmic generation of a consensus-expressing knowledge base. In the next period, the experts continue their parallel work, based on this consensus file. It generates considerably more workload for experts, but the accumulating uncertainty can be decreased.

Another possible solution to decrease the accumulated uncertainty may be forgetting of the very uncertain binary relations. Accordingly, beyond a certain number of evaluation cycles, the binary relations with high uncertainty can be forgotten temporarily or permanently. It can be solved by changing the ‘forget’ flag at the end of `rmin()` relations, signaling that the actual relation is not taken into consideration in the ‘propose’ algorithm.

Assuming that frequently used relations with high average value and low uncertainty are part of highly feasible solutions, the data set was filtered accordingly. 48 binary relations have a high average value ≥ 0.7 , 25 of which have a low uncertainty ≤ 0.3 . Potentially, those 25 pairs are part of highly feasible solutions. Yet, the frequency of occurrence of those 25 binary relations ranges between 2 and 46, only eight relations occur 30 or more times.

Vice versa, it is assumed that frequently used relations with low average value and low uncertainty are part of low-performing solutions. 54 binary relations have a low average value ≤ 0.35 , 14 of which have a low uncertainty ≤ 0.3 . Potentially, those 14 pairs are part of low-performing solutions. Yet, the frequency of occurrence of those 14 binary relations ranges between 1 and 60, only four relations occur 30 or more times.

Indicating a functioning algorithm, the strongest average value of 0.9 and a low uncertainty of 0.2 is attributed to the pair p5_7 Data Mining and Analytics and p6_1 Convolutional Neural Networks. The second strongest average value of 0.75 and an uncertainty of 0.1 is attributed to the pair p1_3 Process control and p5_9 Computer-Aided Manufacturing. Both these pairs are naturally fitting partners. Additional pairs with high average values and low uncertainty underline the conclusion of a functioning algorithm: p1_8 Maintenance and p5_10 Real-time Monitoring and Control Systems, p1_9 Quality Control and p5_7 Data Mining and Analysis, p1_6 Operation Scheduling and p5_11 Simulation-based Optimization.

Analyzing the latest knowledge base, Rmin_DW1TH1SR1full1_7.pl reveals additional interesting findings. As listed in Table 10, filtering property class P1 Task of application, 330 pairs remain out of the total 733 binary relations. 130 of the 330 (39%) are paired with P5 Conventional problem-solving methods. 80 of 330 (24%) are paired with P6 AI/ML-based problem-solving methods. Further analyzing the 25 P1 pairs that reached an Avg Value ≥ 0.7 , 23 of them are paired with P5 properties, and only one pair features a P6 property. This low presence of AI/ML problem-solving methods among the high Avg. value combinations is remarkable. It reflects the evaluations provided by the participating experts and therefore represents the current state of knowledge captured inside the evolving knowledge base. Consequently, the low evaluation of AI/ML-based problem-solving methods may indicate either a perceived limitation of their use in the investigated manufacturing contexts or an actual limitation in the problem-solving ability.

Table 10 Analyzing P1 Task of application in the latest knowledge base

Filter	# of binary relations
none	733
Class 1 P1 Task of application	330
Class 1 P1 Task of application – Class 2 P5 Conventional problem-solving methods	130

Class 1 P1 Task of application – Class 2 P6 AI/ML-based problem-solving methods	80
Class 1 P1 Task of application – Avg value ≥ 0.7	25
Class 1 P1 Task of application – Avg value ≥ 0.7 – Class 2 P5 Conventional problem-solving methods	23
Class 1 P1 Task of application – Avg value ≥ 0.7 – Class 2 P6 AI/ML-based problem-solving methods	1

The high percentage of P1 pairs, shown in table 10, indicates a task-oriented knowledge base. The majority of binaries apply computational problem-solving; the conventional problem-solving methods make up two-thirds. Task orientation, computational problem solving, and the dominance of conventional problem-solving methods are resulting from the expert input.

A total of 280 pairs feature P6 AI/ML-based problem-solving methods; 390 pairs feature P5 Conventional problem-solving methods. Again, we see a dominance of conventional problem-solving. The P5 pairs' average Uncertainty is 0.51 (avg Avg. value 0.50). The P6 pairs average a higher uncertainty average of 0.6 at a comparable average Avg value. The lower Uncertainty average indicates that the P5 pairs provide more impactful new knowledge to the knowledge base. Interestingly, the 84 pairs featuring P5 and P6 properties (excluding pairs with N/A properties) have an even lower average Uncertainty of 0.45, indicating impactful knowledge. Supporting that conclusion, those 84 pairs feature nine pairs ≥ 0.7 Avg value, equaling 10% of binary relations. The complete database with its 733 binaries counts 48 pairs with ≥ 0.7 Avg value, equaling 6.5%. Combining P5 and P6 properties leads to binary combinations, providing the knowledge base with impactful new knowledge.

Assumptions for qualitative conclusions are the following:

- High uncertainty values indicate potentially trivial relations that do not specify new knowledge for the selection of elements while serving as important lessons for the possible reconstruction of the knowledge base.
- Low uncertainty values with feasible frequency specify new knowledge for the selection of elements.

RPS offers unique knowledge representation and fast learning capabilities, resulting from the following features:

- Constructive, equivalence-based classification of the underlying manufacturing tasks and of the corresponding methodologies or methodological combinations.
- Characterization of the relations between properties in these methodological combinations (in graph-theoretic terms, these relations are represented by undirected edges).
- Possibilistic formulation of uncertainty, reflecting the nature of the information involved.
- Coherent representation of missing knowledge, ensuring internal consistency.
- Formal expressions of key characteristics of evaluation distributions, especially those derived from heuristic values proposed by experts.
- Algorithmic construction of aggregation procedures, with optional computation of consensus evaluations across multiple experts.

The main limitation of the implemented methodology is the necessary manpower for expert evaluation. This obviously limits the number of iterative feedback cycles, also in the context of using an automatic consensus, instead of the uncertainty-increasing aggregation. However, this limitation can be avoided if we use the method for another class of problems, when evaluation of complete combinations can be generated based on some simulation-like calculations, automatically (like we used it for the simulated experts in the simplified example in Chapter 5.1). Regardless, in future work, it would be straightforward to apply a stepwise consensus, followed by the in-parallel evaluation, based on this consensus by multiple experts. To decrease manpower, in the knowledge of first evaluations, the combinatorial complexity can be decreased by revision of necessary relations between the property classes and the properties, as was indicated above.

6.3 Interpretation of the learning behavior of the RPS

The results presented and discussed in Chapters 5.1, 5.9, 6.1, and 6.2 demonstrate how the knowledge base evolves during the iterative evaluation process of the RPS. The observed learning reflects the conversion of an initially undefined compatibility structure into a structured representation of uncertain knowledge regarding feasible combinations of computational problem-solving methods for manufacturing tasks.

The initial state of the knowledge base represents a condition of explicit ‘lack of knowledge’. The binary relations between properties are defined only structurally through the compatibility lattice introduced in Chapter 4, without quantitative evaluation. The use of interval values $[m, M]$ to represent relations between properties allows the system to represent possibilistic

uncertainty, explicitly, rather than assuming precise probabilities. These representation of partial knowledge is related to fuzzy and interval-based reasoning approaches (Novák, 2012; Zadeh, 1965), which allows reasoning under incomplete information.

During the evaluation cycles, experts assess complete combinations, and these assessments are propagated downward through the compatibility lattice to update the involved binary relations. This mechanism gradually transforms the initial ‘lack of knowledge’ into structured uncertain knowledge. Similar principles of incorporating expert knowledge into computational learning processes have been discussed in the context of knowledge-informed machine learning and hybrid reasoning systems (J. U. Hansen & Quinon, 2023; Von Rueden et al., 2023). In contrast to many ML approaches that rely on large data sets, the RPS operates primarily on structured expert evaluations and the combinatorial structure of the compatibility lattice.

The results show that the uncertainty associated with binary relations evolves with each iteration of the learning process. Initially, uncertainty decreases as the evaluation of complete combinations provides information about previously unevaluated relations. As the knowledge base becomes more populated, the reduction of ‘lack of knowledge’ slows and the system increasingly represents uncertain knowledge rather than missing information. This transition reflects a shift from ‘lack of knowledge’ toward structured uncertainty, a phenomenon frequently discussed in decision-support systems where limited knowledge must be refined through iterative evaluation processes (Horvath & Rudas, 2004; Nonaka & Takeuchi, 2000).

The convergence of uncertainty from “Lack of knowledge” toward zero requires interpretation. This effect does not imply that the underlying knowledge becomes objectively accurate or universally valid. Rather, it reflects the progressive elimination of the “Unknown” state within the RPS as binary relations are increasingly evaluated through the iterative process. In this sense, the reduction of uncertainty indicates an increasing completeness of the knowledge base, not necessarily an increase in epistemic correctness. Consequently, the framework should be interpreted as identifying tendentially better-performing combinations under the given knowledge conditions, while maintaining a residual and context-dependent uncertainty. This distinction is essential for correctly understanding the learning behavior of the RPS and its role as a proposal-generating methodology rather than a validated decision-support system.

Another important observation is that the algorithm is able to generate new complete combinations that tend to perform better according to expert evaluation. This indicates that the compatibility lattice, together with the feedback mechanism between maximal and minimal

elements, provides a useful representation for exploring large combinatorial spaces of possible task-method combinations. Such combinatorial problems are common in engineering design and manufacturing planning, where different computational tools must be combined to address complex challenges (Groover, 2020; Segreto & Teti, 2014).

The results further illustrate the role of expert knowledge in guiding the exploration of the possibility space. Expert evaluation serves as the primary source of information used to update the knowledge base and guide the generation of new combinations. The importance of domain expertise in guiding computational decision-support systems has been widely emphasized in the literature (J. U. Hansen & Quinon, 2023; Miller et al., 2024). In manufacturing environments in particular, expert knowledge often complements data-driven approaches by providing contextual understanding of process constraints, operational objectives, and feasible conditions.

The aggregation of multiple experts introduces an additional dimension to the learning process. As shown in Chapter 5.9.3, combining evaluations from multiple experts can enrich the knowledge base by incorporating different perspectives on method feasibility. At the same time, beyond a certain number of iterations, the aggregation of evaluations may increase the uncertainty of the binary relations. This effect reflects the natural diversity of expert judgment, which may introduce conflicting evaluations when experts interpret complex combinations differently. Similar challenges have been discussed in the literature on expert knowledge integration and collaborative decision making (Nonaka & Takeuchi, 2000).

From a methodological perspective, the observed learning behavior shows that the RPS approach provides a structured mechanism for accumulating knowledge about possible combinations of computational problem-solving methods applied in manufacturing. While conventional computational tools such as simulation, optimization, and statistical process control provide powerful mechanisms for solving specific problems (Law, 2013; Montgomery, 2019; Wolsey & Nemhauser, 2014), the proposed RPS framework focuses on the higher-level task of organizing and discovering combinations of such methods within a structured possibility space.

This distinction is particularly relevant in the context of modern manufacturing systems, where a growing variety of computational tools must be integrated to address complex production challenges (Beyer & Sendhoff, 2008; Cortes & Vapnik, 2020; Goodfellow et al., 2016). The

RPS framework contributes to this challenge by providing a systematic method for exploring how such tools may be combined and evaluated to solve specific manufacturing tasks.

The RPS provides a learning environment in which experts iteratively evaluate compatibility relations to generate increasingly informative representations of feasible method combinations. This supports the central objective of the dissertation: to introduce and to test a methodology for the systemic categorization and evaluation of conventional and AI/ML applications in manufacturing.

6.4 Comparison with existing method discovery approaches

Chapter 3.4 introduced several well-known approaches for discovering or selecting methods or combinations of methods in complicated settings. These approaches include probabilistic graphical methods such as Bayesian networks, optimization-oriented methods such as genetic algorithms, interactive learning approaches such as active learning, and structured knowledge representation approaches such as knowledge-graph reasoning.

A first group of approaches consists of probabilistic graphical models, including Bayesian networks, Hidden Markov Models, Markov Random Fields, and related approaches (Sucar, 2021). These approaches work with the relationships between variables in a probabilistic manner and facilitate the inferences of unknown states based on observed data. Bayesian updating will allow to improve knowledge about such relationships as new evidence flows in (Oijen, 2024), whereas Bayesian optimization will attempt to smartly decrease the optimal parameter settings with the help of models that rely on probabilities (Yang et al., 2024). These procedures prove to be especially useful in situations where adequate data is available to approximate probability distributions and where predicting or making probabilistic inferences is the aim. But they are optimizing rather than systematically combining possible methods. The RPS does not quantify the probability of optimum solutions but represents compatibility relations between equivalence-classified properties and distinguishes between uncertainty based on evaluation and formal ‘lack of knowledge’.

A second class of methods concerns interactive or sequential learning strategies, such as active learning and reinforcement learning for approach selection. Active learning algorithms allow a learning system to query experts selectively in order to obtain labels for those examples expected to provide the greatest information gain (Tang & Huang, 2021; Tharwat & Schenck, 2023). Reinforcement learning approaches frame the problem as sequential decision-making, where an agent learns to select actions or methodological choices that maximize a cumulative

reward signal through interaction with the environment (Torres et al., 2022; Vairavasundaram et al., 2023). These approaches are typically used in conjunction with specific algorithmic models, such as machine learning algorithms including neural networks, support vector machines, or other data-driven classifiers (Cortes & Vapnik, 2020; Goodfellow et al., 2016). Both approaches aim to improve decision performance by iteratively learning from data or expert interaction. However, these methods presuppose that the learning process leads to an optimal solution. The RPS has a very different goal: instead of being trained on an optimal decision policy, it is building a structured space of possibilities, where viable combinations of methods for solving computational problems and manufacturing tasks are explored and researched in a systematic manner.

The third category of methods is population-based evolutionary optimization, the biggest of them being genetic algorithms (W. Liu et al., 2025; Petrowski & Ben-hamida, 2017; F. Wang et al., 2024). The candidate solutions provided by genetic algorithms are developed in an iterative process of selection, crossover, and mutation. They are useful in searching large search spaces and finding strong-performing candidate solutions. The evolutionary learning techniques generalize these concepts to ongoing optimization issues with real-valued encodings (Beyer & Sendhoff, 2008). In industrial applications, evolutionary methods may be used to optimize parameters of computational models or to search for effective configurations of decision-support tools. However, the learning process in evolutionary algorithms is driven by the evolution of candidate populations, and the algorithm's memory is limited to the population of evaluated solutions. The RPS maintains an explicit knowledge base of compatibility relations between properties. By repeated reevaluation based on expert feedback, this knowledge base accumulates information about relationships between manufacturing tasks and computational problem-solving methods, enabling previously evaluated combinations to influence the generation of new combinations in a systematic way.

A fourth class of approaches includes structured knowledge-representation methods, such as knowledge-graph reasoning (Zeng et al., 2023). Knowledge graphs represent entities and their relationships as graphs and support reasoning and recommendation tasks based on these relationships. Although these methods have common use of structured relational representations, as in the case of the RPS, their main aim is to determine relationships or recommend entities by querying existing graph systems. The RPS differs as it focuses specifically on compatibility relations between equivalence-classified properties representing

manufacturing tasks and computational problem-solving methods, and it updates these relations through iterative expert evaluation.

The approaches discussed in Chapter 3.4 provide powerful mechanisms for prediction, optimization, recommendation, or sequential decision-making. However, they do not address the problem of systemically synthesizing new combinations of computational problem-solving methods for complex manufacturing tasks under conditions of incomplete knowledge. Many of the computational methods discussed in Chapter 4.1. – such as simulation (Law, 2013), optimization models including MILP and MINLP (Grossmann & Trespalacios, 2013; Wolsey & Nemhauser, 2014), or simulation-based optimization (Fu, 2015) – are typically applied individually to specific problem classes. The RPS addresses a different objective: the systematic synthesis of compatible method combinations under explicit representation of uncertainty and ‘lack of knowledge’.

Finally, the distinction between conventional and AI/ML-based computational problem-solving methods, introduced in Chapter 4.1, further highlights the relevance of a synthesis-oriented framework. Manufacturing systems rely more and more on hybrid combinations of conventional and AI/ML-based methods. Determining workable and practical combinations of these various approaches is thus a major methodological issue. The RPS provides a systematic exploration mechanism by modeling both forms of methods in a single compatibility framework and allowing the systematic generation and appraisal of new method combinations.

6.5 Reconnection of results to the literature

This section contains the discussion of the major findings of Chapter 5 and their relation to the scientific literature reviewed in Chapter 3. Instead of the abstract of a single research, it is intended to compare the identified patterns with the aggregate methodological features obtained based on 269 assessed articles (see Chapter 3.2), which serve as the foundation of the equivalence classification and evaluation model.

One of the key discoveries in this dissertation is that combinations based on conventional computational problem-solving approaches perform very well. This can be attributed to the existing literature on manufacturing decision-making, where model-based and deterministic models often find application because of their interpretability and the potential to model system constraints (as explained in Chapter 3.1 and modeled in Chapter 4.1). These approaches are particularly appropriate at an early stage of problem-solving when logical organization and clarity are needed.

The techniques based on AI/ML are evaluated more poorly in a number of combinations studied, especially those that are implemented in settings lacking sufficient data and those where prior knowledge of the conceptual system is necessary. This is in line with the literature on AI applications in manufacturing, which emphasizes data availability, data quality, and explainability as essential success factors (as discussed in Chapter 3.1). Also, expert knowledge is still critical in complex settings, which supports the results of human-in-the-loop and hybrid systems (as mentioned in Chapter 3.3).

The results also indicate that hybrid combinations of conventional and AI/ML-based computational problem-solving may be beneficial in cases where the strengths of the two methodologies are integrated properly. This point aligns with modern literature on integrating data with model-based approaches to enhance the performance of complex systems (as argued in Chapters 3.3 and 4.1).

Lastly, the results can be viewed in connection with the current approaches of method discovery and optimization. According to Chapter 3.4, probabilistic, evolutionary, and optimization algorithms aim at finding the best solutions in stipulated structures. The RPS, as introduced in Chapter 4.2, is an equivalence classification-based (structured) representation of the solution space, which can be effectively used to systematically generate and evaluate methodological combinations when facing uncertainty. The presented trends in Chapter 5 reveal that such a method can identify significant correlations between the characteristics of the problem and the adequacy of the method.

The results do not aim to validate individual combinations in real-world settings, but to provide a structured, knowledge-based exploration of the methodological possibility space. The plausibility and relevance of the proposed framework are backed by the consistency of the observed patterns with the existing results in the literature.

6.6 Practical implications

For the manufacturing industry, the presented methodology provides value for AI adoption projects. The systematic integration of AI/ML-based and conventional methods and their compatibility relation with manufacturing tasks provides a holistic perspective. Leaders and deciders do not have to assess AI/ML methods independent of their compatibility and integrability with conventional problem-solving methods. And with the integrated structured manufacturing levels and tasks, the matching of potential problem-solving methods is based on a comprehensive, holistic methodology for decision support. By providing expert-evaluated, tendentiously better complete combinations, decision makers start out at a significantly smaller

set of potential solutions, the chance of totally irrelevant suggestions declines with each learning iteration. Thus, the trial-and-error phase of AI adoption is reduced, saving resources and time. The framework supports AI adoption by guiding method selection, not by prescribing deployment. It is a decision-support tool, not an automated solution finder.

Implementation in practice would follow a staged process: use-case definition, data readiness, prototyping, controlled piloting, and scaling. Here is a step-by-step implementation roadmap for the adoption of any selected complete combinations suggested by the framework:

Table 11 Implementation roadmap for the adoption of complete combinations (developed based on Haefner et al., 2023; Paleyes, 2026; Sarkar, 2026; Uren & Edwards, 2023)

#	Step	Activity
1	Define target use case and decision boundaries	Identify the specific task to be handled by the selected combination and define success indicators, constraints, and operational decision points
2	Map the selected combination to the business context	Translate the properties into concrete manufacturing objects: data sources, process owners, IT systems, expected results. Check if the selected combination is feasible with the data, equipment, and decision power available.
3	Establish data and system availability	Assess data, data availability, access rights, quality, sampling frequency, and labeling feasibility. Set up data pipelines if necessary.
4	Build a minimum viable prototype	Implement the smallest version of the selected combination that produces an actionable result. Keep model complexity low and prioritize stable integration into existing workflows.
5	Perform offline evaluation and risk screening	Evaluate the prototype against historical or controlled data. Validate performance using defined indicators and perform quality checks,
6	Conduct a pilot project in a controlled environment	In a limited setting (one production line, one product family, or one site), deploy the prototype involving human intervention in the decision-making process. Track results and verify understanding and trust of users.

7	Integrate into standard processes	Formalize ownership, training and maintenance responsibilities, escalation paths, and monitoring. Establish dashboards to display key performance indicators based on agreed update cycles. Involve compliance and quality management for alignment.
8	Deploy at scale and improve continuously	Deploy to additional lines or sites and iterate based on feedback. Leverage the framework to refine method selection and update assessments as objectives and/or conditions change.

Every technology implementation project encounters barriers. For the implementation of computational problem-solving methods, those barriers can be divided into technological, organizational, financial, and cultural barriers (Heimberger et al., 2024). Technological barriers include data quality, data labeling, legacy systems, lack of instrumentation, and IT security limitations. To overcome these obstacles, a limited use case with minimal instrumentation and strict data management can help. The use of APIs for the integration of legacy systems and thorough monitoring are helpful, too. (Ahangar & Farhat, 2025; Paleyes, 2026; C. Wang, 2025)

Organizational obstacles include unclear responsibilities and accountabilities between engineering, IT, and operations, a lack of process integration, or misaligned incentives. A clearly assigned project owner and a comprehensive RACI chart help in this situation. (Heimberger et al., 2024; Paleyes, 2026; Zarour & Alzabut, 2025) Financial barriers include cost uncertainty, diffuse timelines for the return on investment, and competing interests. Recommendable to mitigate those issues are small-scale pilot approaches, base loss factors, and milestone-based investment after value has been proven. (Paleyes, 2026; Zarour & Alzabut, 2025) Cultural obstacles commonly include trust issues, resistance to change, skepticism about AI, or fear of surveillance. Transparent models at the beginning of the project can help mitigate skepticism. Training can help by focusing on decision support rather than replacement. Human involvement and clear communication of boundaries and responsibilities help to break down barriers. (Ahangar & Farhat, 2025; Heimberger et al., 2024)

7. CONCLUSIONS AND RECOMMENDATIONS

The learning knowledge representation, using a reevaluating algorithm over the lattice of equivalence-classified properties, proved to be applicable for the systemic overview of conventional and AI/ML applications in manufacturing in a way that also supports the algorithmic generation of new solutions from the continuously evolving knowledge base. The proposed algorithms are based on the evaluation feedback between the maximal and minimal elements of a compatibility lattice of the underlying properties.

The practical objective of this work was to systematize the available conventional and AI/ML-based computational problem-solving methods that can be applied to solve various tasks in manufacturing industries. Based on the heuristic evaluation of the related literature, we concluded that the combinatorial complexity of the available methods and tools in the context of the tasks to be solved needs a theoretically established systematic approach.

As a first step, we determined six equivalence classes of properties describing the combinations of the methods with tasks (Task of application, Level of application, Phases of problem solving, Development goal, Conventional problem-solving methods, AI/ML-based problem-solving methods). We also determined the binary compatibility relations between these classes, resulting in 15 binary relations between them.

Next, the relations between the elements in the 15 pairs of equivalence classes were examined, and the possible binary compatibility relations between the underlying properties were determined heuristically, resulting in 733 binary relations.

Accordingly, the possibility space, outlined by the binary relations between the classified properties, was generated.

This qualitative knowledge base made it possible to apply an underrepresented intelligent learning methodology for uncertain quantitative determination of the relations between the conventional and intelligent methods for the solution of various tasks in the manufacturing industry. Accordingly, we applied the theoretically established learning feedback between the complete combinations and the binary relations, determining these complete combinations. The essence of the applied feedback is that normalized, real-valued (within $[0, 1]$ interval) evaluation of complete combinations can generate increasing $[m, M]$ sub-intervals (where $0 \leq m \leq M \leq 1$) for the binary relations. The structured set of these sub-intervals contains an uncertain possibilistic knowledge that, by means of the appropriately developed algorithms, can generate tendentially better new complete combinations.

The initial state of knowledge base can be described by the paradox values of $m=1$ and $M=0$, expressing the ‘lack of knowledge’ in sense of the four-valued logic (where the other cases refer to ‘true’, ‘false’, and ‘uncertain’).

The learning started with 300 (and later continued with 50) complete combinations, evaluated by the author, as well as by two collaborating experts. Based on the reevaluated possibility space, new complete combinations were generated, and having evaluated them, we continued the procedure iteratively.

This model (as it is demonstrated by the simplified case study) utilizes the structural knowledge of equivalence classification-based lattice. Also, it relies on experts’ subjective assessments. It was concluded that along these iteratively repeated learning steps, the uncertainty of ‘lack of knowledge’ essentially decreased, while the uncertainty of accumulated knowledge increased, resulting in a summarized uncertainty with a minimum period, that continues with a slow increase. This expresses the fact that uncertainty of ‘lack of knowledge’ is converted to less uncertainty of knowledge, while it makes possible to generate feasible solutions .

The uncertainty can be quantified by the normalized measure introduced in equations (18)-(21). The uncertainty of lack of knowledge was reduced by 100%, as all binary relations were evaluated during the evaluation cycles. Meanwhile, the uncertainty measure of reevaluated binary relations increased stepwise to 62%, and the summarized uncertainty had a minimum at 43%. Accordingly, the lack of knowledge was eliminated completely during the learning process, accompanied by the smaller increase in uncertain knowledge.

Having tested various proposition strategies for the generation of new complete combinations, it was found that

- random selection order from binary relations between the property classes, as well as
- random choice of the applied selection strategies (actually maxmin = maximize minimal; maxmax = maximize maximal; maxave = maximize average; unknown = eliminate ‘lack of knowledge’) from amongst the respective binary relations

performed the best.

It is to be noted that automatic logging of execution makes it possible to correspond to the reproducibility issues, also in the case of random execution.

It was found that the opinion of multiple experts can be taken into consideration by aggregation, i.e. the blind evaluations of multiple experts reevaluated the same knowledge base of binary

relations, subsequently. However, beyond a certain step of iteration, the aggregation increases the uncertainty.

Having studied the knowledge base after some learning steps, the content of the binary relations was analyzed in the context of their contribution to the value of proposed new combinations.

The final conclusion is that the learning algorithm of the RPS is able to determine new and tendentiously better solutions from the quite large space of combinations. The apparently forgotten, underrepresented methodology of reevaluated lattice is very familiar with the studied knowledge discovery problem, where the comprehensive evaluation of alternative solutions (complete combinations) needs all the feasible building elements, while the (ultimately binary) parts of the complete combinations can inherit uncertain knowledge from the solutions they previously participated in.

The discussed method discovery algorithm cannot be validated easily. Complete empirical experimental validation requires the realization and testing of the suggested new combinations. Simulated validation would need the generation of a simulation model for the given combinations and their underlying processes. In principle, it could be performed, but would need a configurable simulation model, equipped with the necessary initial conditions and parameters.

Considering methodological transparency, the source code of the algorithm, and the necessary files, as well as the calculated results, are available in the respective GitHub repository (Weber & Varga, 2025b).

In future work, for the evaluation of multiple experts, the explicit application of consent and conflict is recommended and planned. In this case, the individual experts evolve their own knowledge bases in parallel through multiple iteration rounds. Afterward, the program can generate the knowledge base of consensus (in the form of intersections of intervals) and of conflict (in the form of (aggregation—consensus)) from the expert-specific knowledge bases. Next, each expert can continue the work with the generated consensus file.

This dissertation leverages expert evaluation to ensure interpretability and methodological transparency. Recent advances in natural language processing and large language models could enable automated or semi-automated expert support in future extensions of the framework.

Lastly, based on the content of the knowledge base of binary relations, further suggestions can be derived for the possible modifications of property classes, properties, and binary relations.

8. NEW SCIENTIFIC RESULTS

Novel contributions of this dissertation are summarized as follows:

- Structured representation of conventional and AI/ML-based computational problem-solving methods, applicable in manufacturing industries, achieved through the implementation and application of a lattice of binary compatibility relations amongst equivalence-classified properties;
- Implementation and application of a Reevaluated Possibility Space for the systematic assessment and discovery of methodological combinations of conventional and AI/ML-based computational problem-solving approaches in manufacturing contexts;
- Analysis of possibilistic uncertainties capturing both ‘lack of knowledge’ and existing (already known) knowledge during the learning process, resulted from the information feedback between expert evaluations and the continuously reevaluated knowledge base of binary relations;
- Comprehensive analysis and discussion of results derived from a detailed case study, focusing on property combinations that characterize task features, as well as conventional and AI/ML-based problem-solving methods, in context of published approaches and existing solutions.

The respective new scientific results are the following:

1. Based on a systematic literature review, relevant conventional and AI/ML-based computational problem-solving methods applicable in manufacturing were identified. From this foundation, six property equivalence classes were defined (Task of application; Level of application; Phase of problem solving; Development goal; Conventional computational problem-solving method; and AI/ML-based computational problem-solving method), and the respective properties were classified accordingly. Using the theoretical basis of the compatibility structures, the lattice amongst the binary compatibility relations of equivalence-classified properties was implemented and applied for the representation of the task-describing properties, as well as the conventional and AI/ML-based computational problem-solving methods in manufacturing industries.

2. A Reevaluable Possibility Space was implemented and applied for systematized assessment and discovery of possible methodological combinations of conventional and AI/ML-based computational problem-solving methods in manufacturing industries. The learning algorithm is based on the evaluation feedback between the evaluated maximal (complete) property

combinations and the cyclically reevaluated minimal (binary) relations characterizing the compatible pairs of classified properties. Guided by the literature and expert knowledge, a first set of complete combinations was created and evaluated by three experts with feasibility values in the range $[0, 1]$. Next, the respective binary knowledge bases were reevaluated by the 'reevaluate' algorithm, resulting in uncertain $[\text{Min}, \text{Max}]$ subintervals within the $[0, 1]$ interval. These reevaluated knowledge bases were aggregated, and new complete combinations were generated from the resulted knowledge base by the 'propose' algorithm for evaluation by the experts. The evaluation feedback realized by the cyclically repeated reevaluations and propositions, resulted in the systematic exploration of methodological combinations.

3. Based on the results from a simplified fictitious example with simulated experts, as well as from the detailed case study, possibilistic uncertainties characterizing 'lack of knowledge' and already known knowledge during the learning process were analyzed. The main results were as follows:

- The Reevaluable Possibility Space resulted in fast elimination of 'lack of knowledge', accompanied by the increase of uncertain knowledge in all studied cases.
- Randomization in a simplified example illustrated that the essential functioning of Reevaluable Possibility Space is not affected by the number of evaluating experts.
- The increasing possibilistic uncertainty of the already evaluated binary relations consists of two parts. The major part comes from the inherent uncertainty, expressing the situation that a single binary relation can participate in multiple complete combinations with different values. The second part expresses the possible erroneous and/or subjective evaluations of the experts. This can be higher as the number of experts increases, as it incorporates their diverse experiences, while aggregation of evaluations will be affected by the more uncertain expert opinions.
- A cyclically generated consensus, followed by continuation of evaluation feedback of each expert, can cyclically decrease the uncertainty.
- The comparison of the results from the simplified and the detailed real-world case studies underlines the scalability of the method.

4. The results, obtained from the detailed case study about property combinations representing the features of the tasks to be solved, as well as the conventional and AI/ML-based problem-solving methods in manufacturing were analyzed and discussed in the context of published methods and individual solutions in the literature. It was concluded that the algorithmic learning process of Reevaluable Possibility Space provided practically relevant insights into

methodological combinations in manufacturing. Successful solutions often involved both conventional computational techniques and AI/ML-based ones, but not only AI/ML containing methods. The evolving knowledge base enabled the identification of advantageous and disadvantageous solutions, contributing to a structured understanding of feasible methodological combinations. Frequently appearing binary combinations of high uncertainty highlights the optional elimination of these relations.

9. SUMMARY

There is a rapid change in manufacturing systems due to the growing complexity, dynamic demand of the market, and sustainability requirements. Artificial Intelligence (AI) and Machine Learning (ML) present positive opportunities to enhance the efficiency of processes, quality control and quality assurance, supply chain resilience, and resource optimization. Yet, although different applications are used, they are often considered separately in literature, without a systematic approach to learning their interconnections or value-added.

The proposed systematic approach to aggregating, assessing, and refining sets of conventional and AI/ML-based computational algorithms to solve advanced manufacturing issues addresses this gap. It uses expert knowledge, reevaluable lattices, and learning algorithms to build and enhance an organized body of knowledge that can be evolved, iteratively.

The main objective of this work was to introduce and to test a methodology for the systemic categorization and evaluation of conventional and AI/ML-based applications in manufacturing, in a way that supports a clear overview of the existing methodological combinations, as well as contributes to the automatic generation-based discovery of new solutions from a continuously evolving knowledge base of the existing methods. The specific aims were the following:

- to classify the properties, characterizing the engineering and management tasks, as well as the applied conventional and AI/ML-based methods and tools used for developing sustainable manufacturing;
- to assess the usefulness of computer-supported problem solving in manufacturing, based on published case studies and experts' knowledge;
- to create a learning knowledge base for the development of usable, advantageously AI-involved, optionally combined computational methods and tools for supporting the development of new, effective, and sustainable (e.g., circular) solutions.

The core structure of the approach relies on six equivalence classes of:

- P1 – Task of Application
- P2 – Level of Application
- P3 – Phases of Problem Solving
- P4 – Development Goal
- P5 – Conventional Computational Methods
- P6 – AI/ML-based Computational Methods

A total of 15 binary relations were established between these classes, resulting in 733 compatibility relations derived heuristically. These formed a compatibility lattice, supporting the implementation of a Reevaluable Possibility Space (RPS).

An evaluation feedback-driven learning algorithm processes information to evolve a knowledge base, while reducing uncertainty related to ‘lack of knowledge’ following this sequence:

- Initial combinations (maximal relations) were constructed using one element from each class.
- These were evaluated by experts using real-valued scores in $[0, 1]$ interval.
- Evaluations informed updates to the binary compatibility relations, expressed as $[\text{Min}, \text{Max}]$ sub-intervals.
- The algorithm proposed new combinations tendentially better than previous ones.
- Iterative feedback cycles refined the knowledge base, reducing uncertainty related to ‘lack of knowledge’ and mapping more promising solution paths.
- Evaluations were conducted by the author and two collaborating experts.
- Aggregated expert knowledge was incorporated consistently.
- Random strategies (e.g., random selection of binary relations or scoring methods like maxmin, maxave) performed best in generating effective new combinations.

Practical Insights:

Analyzing the 50 newly suggested and evaluated maximal combinations of the latest iteration (Rmax_DW1TH1SR1full1_650), the following practical insights can be derived.

- AI/ML-based problem-solving methods are not deemed advantageous in the knowledge base at this stage, given a disproportionate occurrence of maximal combinations without an AI/ML-based solution approach (property p6_8 N/A). This observed result reflects the evaluations provided by the participating experts and therefore represents the current state of knowledge captured inside the evolving knowledge base. Consequently, the low evaluation of AI/ML-based problem-solving methods may indicate either a perceived limitation of their use in the investigated manufacturing contexts or an actual limitation in the problem-solving ability.
- Convolutional Neural Networks (CNNs) – despite being widely discussed – have limited practical relevance in current manufacturing settings. They only occur once in the top ten-evaluated combinations.

- Operation scheduling (property p1_6) is deemed an extraordinarily advantageous task to be solved, while the Process unit (property p2_1) is the least advantageous level within the focused complete combinations. The evolving uncertainty landscape reveals that total uncertainty reaches a minimum during intermediate learning stages, after which it gradually increases due to accumulating complex knowledge.

The novel contributions of this dissertation are summarized as follows:

- A tested methodology for the systematic categorization and evaluation of conventional and AI/ML-based applications in manufacturing
- A structured representation of methodological combinations
- An evolving and learning-capable knowledge base
- The automated generation of new solutions based on this knowledge base

Recommendations for Future Work:

1. Modeling expert consensus and conflict: Apply the explicit models for handling agreement and disagreement between experts, using intersections and differences of [Min, Max] intervals.
2. Temporary forgetting mechanisms: Introduce the option to disregard highly uncertain binary relations to manage complexity.
3. Dynamic refinement of knowledge base: Use insights from the evolving relations to adapt the structure of equivalence classes and the methods they contain.
4. Automation of complete combination evaluation:
 - Leveraging Natural Language Processing and/or Large Language Models to generate or co-generate evaluations.
 - Assess the combination of the RPS with dynamic simulation applications to allow simulation-based evaluation of complete combinations.

Conclusion

This research demonstrates that by combining equivalence classification, expert knowledge, and a learning algorithm, the proposed methodology offers both theoretical lessons and practical suggestions for navigating the growing complexity of industrial problem-solving. The dissertation contributes to a deeper understanding of method integration and opens new pathways for intelligent decision support in manufacturing environments.

10. ACKNOWLEDGEMENTS

I acknowledge the valuable contribution of Thomas Hohnloser (Chief Information Officer, Infracore Wiesbaden GmbH & Co. KG) and Steffen Robus (IT project leader, Deka Bank Deutsche Girozentrale) during the evaluations.

11. DECLARATION

I hereby affirm that I am the author of this dissertation and that I have not utilized any supplementary materials other than those listed.

I hereby declare that this dissertation is my original work, prepared after registering for the Ph.D. degree at the Hungarian University of Agriculture and Life Sciences, and that it has not been previously included or submitted in any work for a degree, diploma, or other qualification at this or any other institution.

Corning, New York, United States of America, 20. April 2026

Dennis Weber

12. APPENDICES

A1: Bibliography

- Ahangar, M. N., & Farhat, Z. A. (2025). AI Trustworthiness in Manufacturing: Challenges, Toolkits, and the Path to Industry 5.0. *Sensors*, 25(14).
<https://doi.org/https://doi.org/10.3390/s25144357>
- Alfere, S. S., & Maghari, A. Y. (2018). Prediction of Student's Performance Using Modified KNN Classifiers. *The First International Conference on Engineering and Future Technology (ICEFT 2018)*, *Iceft*, 143–150.
- Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.
- April, J., Glover, F., Kelly, J. P., & Laguna, M. (2003). Practical introduction to simulation optimization. *Proceedings of the 2003 Winter Simulation Conference, 2003*.
<https://doi.org/10.1109/WSC.2003.1261410>
- Arva, P., & Csukas, B. (1989). Synthesis of engineering objects by recursive fuzzy valuation of crisp combinations. *Fuzzy Sets Ans Systems*, 32, 13–33. [https://doi.org/10.1016/0165-0114\(89\)90085-7](https://doi.org/10.1016/0165-0114(89)90085-7)
- Ben-Eli, M. U. (2018). Sustainability: definition and five core principles, a systems perspective. *Sustainability Science*, 13(5), 1337–1343. <https://doi.org/10.1007/s11625-018-0564-3>
- Bertsekas, D. P. (2019). *Reinforcement Learning and Optimal Control*. Athena Scientific, Belmont, Massachusetts.
- Beyer, H. G., & Sendhoff, B. (2008). Covariance matrix adaptation revisited - The CMSA evolution strategy. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5199 LNCS, 123–132. https://doi.org/10.1007/978-3-540-87700-4_13
- Blickle, T. (ed. . (1977). *Anyag- és hőátadási rendszerek matematikai modelljei (Mathematical Models of Mass and Heat Transferring Processes)*. Műszaki Könyvkiadó.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122. <https://doi.org/10.1561/22000000016>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5–32.
<https://link.springer.com/article/10.1023/a:1010933404324>
- Breyfogle, F. W. (2003). Implementing Six Sigma: Smarter Solutions Using Statistical Methods. In *Technometrics* (Vol. 42, Issue 3). John Wiley & Sons.
<https://doi.org/10.2307/1271088>
- Brynjolfsson, E., & McAfee, A. (2014). *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton & Company.
- Business Roundtable. (2019). *Statement on the purpose of a corporation*.
https://system.businessroundtable.org/app/uploads/sites/5/2023/02/WSJ_BRT_POC_Ad.pdf

- Buttazzo, G. C. (2011). *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications* (Third). Springer. <https://doi.org/10.1007/978-1-4614-0676-1>
- Cavalcante, I. M., Frazzon, E. M., Forcellini, F. A., & Ivanov, D. (2019). A supervised machine learning approach to data-driven simulation of resilient supplier selection in digital manufacturing. *International Journal of Information Management*, 49(March), 86–97. <https://doi.org/10.1016/j.ijinfomgt.2019.03.004>
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3). <https://doi.org/10.1145/1961189.1961199>
- Chen, F. C., Tzeng, Y. F., Hsu, M. H., & Chen, W. R. (2010). Combining Taguchi method, principal component analysis and fuzzy logic to the tolerance design of a dual-purpose six-bar mechanism. *Transactions of the Canadian Society for Mechanical Engineering*, 34(2), 277–293. <https://doi.org/10.1139/tcsme-2010-0017>
- Chen, P., Chu, Z., & Zhao, M. (2024). The Road to corporate sustainability: The importance of artificial intelligence. *Technology in Society*, 76(July 2023), 102440. <https://doi.org/10.1016/j.techsoc.2023.102440>
- Chowdhary, K. R. (2020). First order Predicate Logic. *Fundamentals of Artificial Intelligence*.
- Conklin, T. (2019). *Pre-Accident Investigations*. Taylor & Francis. <https://doi.org/10.4324/978131524617>
- Cortes, C., & Vapnik, V. N. (2020). Support-vector machines. *Machine Learning*, 20, 273–297. <https://doi.org/10.1007/BF00994018>
- Csukas, B., Kozar, Z., & Arva, P. (1989). Multicriteria valuated prolog synthesizing algorithms. *Computers Chem. Enineering*, 13(415), 595–602. [https://doi.org/10.1016/0098-1354\(89\)85043-4](https://doi.org/10.1016/0098-1354(89)85043-4)
- Csukás, B., Varga, M., Balogh, S., Miskolczi, N., Angyal, A., Bartha, L., Szakács, H., & Varga, C. (2012). Knowledge based model for polymer composite design and production. *Materials and Design*, 38, 74–90. <https://doi.org/10.1016/j.matdes.2012.01.041>
- Dadios, E. (2012). *Fuzzy logic: controls, concepts, theories, and applications* (IntechOpen (ed.)). BoD - Books on Demand.
- Davenport, T. H., Harris, J. G., & Morison, R. (2010). *Analytics at Work - Smarter Decisions, Better Results*. Harvard Business School Publishing Corporation.
- Dubois, D., Nguyen, H. T., & Prade, H. (2000). Possibility Theory, Probability and Fuzzy Sets Misunderstandings, Bridges and Gaps. In D. Dubois & H. Prade (Eds.), *Fundamentals of Fuzzy Sets. The Handbooks of Fuzzy Sets Series* (pp. 343–438). Springer. https://doi.org/10.1007/978-1-4615-4429-6_8
- Dudley, J. J., & Kristensson, P. O. (2018). A Review of User Interface Design for. *ACM Transactions on Interactive Intelligent Systems*, 8(2).
- Ebert, R. J., & Griffin, R. W. (2016). *Business essentials* (12 (Januar). Pearson.

- Elkington, J. (1997). Triple bottom line. *Environmental Quality Management*, 8(1), 37–51.
- Encarnacao, J. L., Lindner, R., & Schlechtendahl, E. G. (2012). *Computer Aided Design: Fundamentals and System Architectures* (Second edi). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-84056-2>
- Eslami, Y., Dassisti, M., Lezoche, M., & Panetto, H. (2019). A survey on sustainability in manufacturing organisations: dimensions and future insights. *International Journal of Production Research*, 57(15–16), 5194–5214. <https://doi.org/10.1080/00207543.2018.1544723>
- Fu, M. C. (2015). *Handbook of simulation optimization*. Springer.
- Geissdoerfer, M., Savaget, P., Bocken, N. M. P., & Hultink, E. J. (2017). The Circular Economy – A new sustainability paradigm? *Journal of Cleaner Production*, 143, 757–768. <https://doi.org/10.1016/j.jclepro.2016.12.048>
- Giarratano, J. C., & Gary D. Riley. (2004). *Expert Systems: Principles and Programming* (4th ed.). Course Technology.
- Gockenbach, M. S. (2006). *Understanding and Implementing the Finite Element Method*. Society for Industrial and Applied Mathematics.
- Goldberg, D. E. (2013). *The Design of Innovation - Lessons from and for Competent Genetic Algorithms*. <https://doi.org/10.1007/978-1-4757-3643-4>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT press.
- Grätzer, G. (2003). *General Lattice Theory*. Birkhäuser.
- Groover, M. P. (2020). *Fundamentals of Modern Manufacturing* (Seventh ed). John Wiley & Sons.
- Grossmann, I. E., & Trespalacios, F. (2013). Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE Journal*, 59(9), 3276–3295. <https://doi.org/10.1002/aic.14088>
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354–377. <https://doi.org/10.1016/j.patcog.2017.10.013>
- Haefner, N., Parida, V., Gassmann, O., & Wincent, J. (2023). Technological Forecasting & Social Change Implementing and scaling artificial intelligence : A review , framework , and research agenda. *Technological Forecasting & Social Change*, 197(June 2022), 122878. <https://doi.org/10.1016/j.techfore.2023.122878>
- Han, J., Pei, J., & Tong, H. (2023). *Data Mining - Concepts and Techniques* (Fourth edi). Morgan Kaufman.
- Hansen, J. U., & Quinon, P. (2023). The importance of expert knowledge in big data and machine learning. *Synthese*, 201(2), 1–21. <https://doi.org/10.1007/s11229-023-04041-5>
- Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1), 1–18. <https://doi.org/10.1162/106365603321828970>

- Heimberger, H., Horvat, D., & Schultmann, F. (2024). Exploring the factors driving AI adoption in production : a systematic literature review and future research agenda. *Information Technology and Management*, 0123456789. <https://doi.org/10.1007/s10799-024-00436-z>
- Ho, T. K. (1995). Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 278–282. <https://doi.org/10.1109/ICDAR.1995.598994>
- Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. *SIAM Review*, 18(3), 529–530. <https://doi.org/10.7551/mitpress/1090.001.0001>
- Horvath, L., & Rudas, I. J. (2004). *Modeling and Problem Solving Techniques for Engineers*. Elsevier.
- Huggins, R., & Izushi, H. (2011). *Competition, competitive advantage and clusters: the ideas of Michael Porter*. Oxford University Press.
- Hughes, T. J. R. (2012). *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Inc.
- ISO. (2018). *ISO-45001-2018* (p. 52). International Organization for Standardization. <https://www.iso.org/standard/63787.html>
- Jackson, P. M. (1998). *Introduction to Expert Systems* (Third edit). Addison-Wesley.
- Jong, K. De. (2017). Evolutionary Computation : A Unified Approach. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 373–388. <https://doi.org/10.1145/3067695.3067715>
- Kakaomerlioglu, D. C., & Carlsson, B. (1999). Manufacturing in decline? A matter of definition. *Economics of Innovation and New Technology*, 8(3), 175–196. <https://doi.org/10.1080/104385999000000008>
- Kar, A. K., Choudhary, S. K., & Singh, V. K. (2022). How can artificial intelligence impact sustainability: A systematic literature review. *JOURNAL OF CLEANER PRODUCTION*, 376. <https://doi.org/10.1016/j.jclepro.2022.134120>
- Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (12th ed.). John Wiley & Sons.
- Khakurel, J., Penzenstadler, B., Porras, J., Knutas, A., & Zhang, W. (2018). The Rise of Artificial Intelligence under the Lens of Sustainability. *Technologies*, 6(4), 1–18. <https://doi.org/10.3390/technologies6040100>
- Kirchherr, J., Reike, D., & Hekkert, M. (2017). Conceptualizing the circular economy: An analysis of 114 definitions. *Resources, Conservation and Recycling*, 127(April), 221–232. <https://doi.org/10.1016/j.resconrec.2017.09.005>
- Krajnc, D., & Glavič, P. (2005). A model for integrated assessment of sustainable development. *Resources, Conservation and Recycling*, 43(2), 189–208. <https://doi.org/10.1016/j.resconrec.2004.06.002>
- Kühl, N., Schemmer, M., Goutier, M., & Satzger, G. (2022). Artificial intelligence and machine learning. *Electronic Markets*, 32(4), 2235–2244.

<https://doi.org/10.1007/s12525-022-00598-0>

- Law, A. M. (2013). *Simulation Modeling and Analysis*. McGraw Hill Education.
- Leo Kumar, S. P. (2019). Knowledge-based expert system in manufacturing planning: state-of-the-art review. *International Journal of Production Research*, 57(15–16), 4766–4790. <https://doi.org/10.1080/00207543.2018.1424372>
- Levinson, M. (2017). What Is Manufacturing ? Why Does the Definition Matter ? In *Congressional Research Service Report* (Vols. 7–5700, Issue R44755). www.crs.gov
- Leyffer, S. (2001). Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18(3), 295–309. <https://doi.org/10.1023/A:1011241421041>
- Li, C., Chen, Y., & Shang, Y. (2022). A review of industrial big data for decision making in intelligent manufacturing. *Engineering Science and Technology, an International Journal*, 29, 101021. <https://doi.org/10.1016/j.jestch.2021.06.001>
- Liu, J. W. S. (2000). *Real-Time Systems* (First). Prentice Hall.
- Liu, W., Kuang, Z., Zhang, Y., & Zhou, B. (2025). An effective hybrid genetic algorithm for the multi-robot task allocation problem with limited span. *Expert Systems With Applications*, 280(March), 127–299. <https://doi.org/10.1016/j.eswa.2025.127299>
- Luger, G. F., & Stubblefield, W. A. (1989). Artificial Intelligence and the Design of Expert Systems. *The Benjamin-Cummings Series in Artificial Intelligence*.
- Mansdorf, S. Z. (2019). *Handbook of Occupational Safety and Health*. Wiley & Sons. <https://doi.org/10.1002/9781119581482>
- McCullogh, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin for Mathematical Biophysics*, 5, 115–133. <https://doi.org/10.1007/BF02478259>
- Michael E. Porter. (1998). *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press.
- Michigan State University - Global EDGE. (2000). Industrial Manufacturing. In *Industrial Manufacturing*. <https://doi.org/10.1201/9781482283273>
- Miller, T., Durlík, I., Łobodzińska, A., Dorobczyński, L., & Jasionowski, R. (2024). AI in Context: Harnessing Domain Knowledge for Smarter Machine Learning. *Applied Sciences (Switzerland)*, 14(24). <https://doi.org/10.3390/app142411612>
- Mitchell, M. (1998). *An introduction to Genetic Algorithms*. MIT press.
- Montgomery, D. C. (2019). *Introduction to Statistical Quality Control*. John Wiley & Sons.
- Moore, J. E., Mascarenhas, A., Bain, J., & Straus, S. E. (2017). Developing a comprehensive definition of sustainability. *Implementation Science*, 12(1), 1–8. <https://doi.org/10.1186/s13012-017-0637-1>
- Naser, M. Z. (2025). Why does machine learning work really well in many engineering problems? *Machine Learning for Computational Science and Engineering*, 1(30), 1–13. <https://doi.org/10.1007/s44379-025-00032-0>

- National Safety Council. (2015). *Accident prevention manual for Business & Industry: Engineering and Technology* (14th ed.). National Safety Council.
- Nonaka, I., & Takeuchi, H. (2000). Classic Work: Theory of Organizational Knowledge Creation. In *Knowledge Management Classic and Contemporary Work* (pp. 139–182). MIT press.
- Novák, V. (2012). Reasoning about mathematical fuzzy logic and its future. *Fuzzy Sets and Systems*, 192, 25–44. <https://doi.org/10.1016/j.fss.2010.09.019>
- Oblak, L., Kuzman, M. K., & Grošelj, P. (2017). A fuzzy logic-based model for analysis and evaluation of services in a Manufacturing company. *Journal of Applied Engineering Science*, 15(3), 258–271. <https://doi.org/10.5937/jaes15-13399>
- Occupational Safety and Health Administration (OSHA). (2015). General industry digest. *Osha 2201-08R*, 1–105.
- Oduguwa, V., Tiwari, A., & Roy, R. (2005). Evolutionary computing in manufacturing industry: An overview of recent applications. *Applied Soft Computing Journal*, 5(3 SPEC.ISS.), 281–299. <https://doi.org/10.1016/j.asoc.2004.08.003>
- Oijen, M. van. (2024). *Bayesian Compendium*. Springer International Publishing. <https://doi.org/10.1007/978-3-031-66085-6>
- Olawade, D. B., Wada, O. Z., David-Olawade, A. C., Fapohunda, O., Ige, A. O., & Ling, J. (2024). Artificial intelligence potential for net zero sustainability: Current evidence and prospects. *Next Sustainability*, 4(February), 100041. <https://doi.org/10.1016/j.nxsust.2024.100041>
- Paleyes, A. (2026). *Challenges in Deploying Machine Learning : A Survey of Case Studies*. 55(6). <https://doi.org/10.1145/3533378>
- Petrowski, A., & Ben-hamida, S. (2017). *Evolutionary Algorithms* (Vol. 9). John Wiley & Sons. <https://doi.org/10.1002/9781119136378>
- Pisano, G. P., & Shih, W. C. (2009). Restoring american competitiveness. *Harvard Business Review*, 87(7).
- Plastino, E., & Purdy, M. (2018). Game changing value from Artificial Intelligence: eight strategies. *Strategy & Leadership*, 46(1), 16–22. <https://doi.org/10.1108/SL-11-2017-0106>
- Plathottam, S. J., Rzonca, A., Lakhnori, R., & Illoeje, C. O. (2023). A review of artificial intelligence applications in manufacturing operations. *Journal of Advanced Manufacturing and Processing*, 5(3), 1–19. <https://doi.org/10.1002/amp2.10159>
- Quesada, I., & Grossmann, I. E. (1992). An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16(10–11), 937–947. [https://doi.org/10.1016/0098-1354\(92\)80028-8](https://doi.org/10.1016/0098-1354(92)80028-8)
- Ross, T. (2009). *Fuzzy logic with engineering applications*. John Wiley & Sons.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211–

252. <https://doi.org/10.1007/s11263-015-0816-y>

Russel, S., & Norvig, P. (2020). *Artificial intelligence: a modern approach* (4th ed.). Pearson.

Sarkar, A. (2026). From AI Hype to Workflow Reality: A Strategic Framework for Integrating Generative AI Across Organizational Functions. *Organizational Dynamics*, 55(1), 101202. <https://doi.org/10.1016/j.orgdyn.2025.101202>

Schilling, M. A. (2022). *Strategic management of technological innovation* (7th ed.). McGraw Hill.

Segreto, T., & Teti, R. (2014). Manufacturing. In L. Laperrière & G. Reinhart (Eds.), *CIRP Encyclopedia of Production Engineering* (pp. 828–830). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20617-7_6561

Sharma, V., Sharma, V., & Shukla, O. J. (2023). *Principles and Practices of CAD/CAM*. CRC Press. <https://doi.org/10.1201/9781003350842>

Sucar, L. E. (2021). *Probabilistic Graphical Models: Principles and Applications*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-61943-5>

Tanaka, K., Ikeda, T., & Wang, H. O. (1998). Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs. *IEEE Transactions on Fuzzy Systems*, 6(2), 250–265. <https://ieeexplore.ieee.org/abstract/document/669023>

Tang, Y., & Huang, S. (2021). Dual Active Learning for Both Model and Data Selection. *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, 62076128, 3052–3058.

Tharwat, A., & Schenck, W. (2023). A Survey on Active Learning: State-of-the-Art, Practical Challenges and Research Directions. *Mathematics*, 11(4), 820. <https://doi.org/10.3390/math11040820>

Thiebes, S., Lins, S., & Sunyaev, A. (2021). Trustworthy artificial intelligence. *Electronic Markets*, 31(2), 447–464. <https://doi.org/10.1007/s12525-020-00441-4>

Tobin, K. W., Gleason, S. S., & Karnowski, T. P. (1998). Adaptation of the fuzzy k-nearest neighbor classifier for manufacturing automation. *Proceedings of SPIE - The International Society for Optical Engineering*, 3306, 122–130. <http://dx.doi.org/10.1117/12.301238>

Torres, R., Andreiana, D. S., Ojeda, Á., Hern, A., Enrique, L., & Galicia, A. (2022). A Review of Deep Reinforcement Learning Approaches for Smart Manufacturing in Industry 4 . 0 and 5 . 0 Framework. *Applied Sciences (MDPI)*, 12(23).

Uren, V., & Edwards, J. S. (2023). Technology readiness and the organizational journey towards AI adoption: An empirical study. *International Journal of Information Management*, 68(March 2022), 102588. <https://doi.org/10.1016/j.ijinfomgt.2022.102588>

Vairavasundaram, S., Vairavasundaram, I., Sivamayil, K., Rajasekar, E., Aljafari, B., & Nikolovski, S. (2023). A Systematic Study on Reinforcement Learning Based Applications. *Energies (MDPI)*, 16(3), 1–23. <https://doi.org/10.3390/en16031512>

Varga, M., Csukas, B., Khanal, S., & Bakshi, B. R. (2023). Lessons from the biosphere for the anthroposphere: Analysis of recycling structures of conservational measures.

- Resources, Conservation and Recycling*, 192(October 2022), 106919.
<https://doi.org/10.1016/j.resconrec.2023.106919>
- Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., & Schuecker, J. (2023). Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1), 614–633. <https://doi.org/10.1109/TKDE.2021.3079836>
- Wang, C. (2025). Machine Learning for Industrial Optimization and Predictive Control: A Patent-Based Perspective with a Focus on Taiwan's High-Tech Manufacturing. *Processes*, 13(7), 1–27. <https://doi.org/https://doi.org/10.3390/pr13072256>
- Wang, F., Zhao, L., & Kee, J. (2024). Modeling and solving time-sensitive task allocation for USVs with mixed capabilities. *Ocean Engineering*, 313(P3).
<https://doi.org/10.1016/j.oceaneng.2024.119614>
- Weber, D., & Varga, M. (2025a). A Reevaluable Property Lattice-Based Knowledge Representation for Proposing and Assessing Computational Tools in Manufacturing. *Machine Learning and Knowledge Extraction*, 7(4), 161.
<https://doi.org/10.3390/make7040161>
- Weber, D., & Varga, M. (2025b). *RPS_algorithm*.
https://github.com/monikavarga/RPS_algorithm
- Wieland, R., & Mirschel, W. (2017). Combining expert knowledge with machine learning on the basis of fuzzy training. *Ecological Informatics*, 38, 26–30.
<https://doi.org/10.1016/j.ecoinf.2017.01.001>
- Wolsey, L. A., & Nemhauser, G. L. (2014). *Integer and Combinatorial Optimization*. John Wiley & Sons.
- Yang, K., Liu, L., & Wen, Y. (2024). The impact of Bayesian optimization on feature selection. *Scientific Reports*, 14(3948), 1–11. <https://doi.org/10.1038/s41598-024-54515-w>
- Yin, S., & Kaynak, O. (2015). Big Data for Modern Industry: Challenges and Trends. *Proceedings of the IEEE*, 103(2), 143–146.
<https://doi.org/10.1109/JPROC.2015.2388958>
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8, 338–353.
[https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1), 3–28. [https://doi.org/10.1016/0165-0114\(78\)90029-5](https://doi.org/10.1016/0165-0114(78)90029-5)
- Zarour, M., & Alzabut, H. (2025). MLOps best practices , challenges and maturity models : A systematic literature review. *Information and Software Technology*, 183(October 2024), 107733. <https://doi.org/10.1016/j.infsof.2025.107733>
- Zeng, Z., Cheng, Q., & Si, Y. (2023). Logical Rule-Based Knowledge Graph Reasoning : A Comprehensive Survey. *Mathematics (MDPI)*, 11(21).
<https://doi.org/10.3390/math11214486>
- Zheng, Y., Yang, S., & Cheng, H. (2019). An application framework of digital twin and its

case study. *Journal of Ambient Intelligence and Humanized Computing*, 10(3), 1141–1153. <https://doi.org/10.1007/s12652-018-0911-3>

A2: SWI-Prolog code of the simplified implementation of the RPS algorithm

```

:- dynamic task/5,rmax/5,rmin/11,selr/5.
:- discontinuoustask/5,rmax/5,rmin/11,selr/5.

:- initialization(run,main).

run :-
    autoload_all,

    retractall(rmax(_,_,_,_,_)),
    retractall(rmin(_,_,_,_,_,_,_,_,_,_)),

    format(atom(Task),'DW~cTask.pl',[92]),
    consult(Task),
    prepare,! .

prepare :-
    task(Program,From1,From2,To,LogName,Input_parameters),
    format(atom(Log),'DW~c~a.txt',[92,LogName]),
    open(Log,write,RL,[buffering(none)]),
    format(atom(Title),'Task: ~a',[Program]),
    write(RL,Title),
    nl(RL),nl(RL),
    format(atom(Input1),'DW~c~a.pl',[92,From1]),
    format(atom(File1),'First input file: ~a',[Input1]),
    write(RL,File1),
    nl(RL),
    consult(Input1),
    format(atom(Input2),'DW~c~a.pl',[92,From2]),
    format(atom(File2),'Second input file: ~a',[Input2]),
    write(RL,File2),
    nl(RL),
    consult(Input2),

    fut(Program,To,Input_parameters,RL),
    close(RL),!.

/*****/

fut(reevaluate,To,Input_parameters,RL) :-
    format(atom(Output),'DW~c~a.pl',[92,To]),
    format(atom(File3),'Output file: ~a',[Output]),
    write(RL,File3),
    nl(RL),
    open(Output,write,RF,[buffering(none)]),
    reevaluate(RL),

    final_state_min(RF),

    findall(Unc,rmin(_,_,_,_,_,_,_,_),Unc,_),Unclist),
    findall(Unc0,rmin(_,_,_,_,_,_,_,_,1,0,Unc0,_),Unc0list),

    sum_of_list(Unclist,0,Sum),
    length(Unclist,Nu),

    sum_of_list(Unc0list,0,Sum0),
    length(Unc0list,Nu0),

    Nuok is Nu-Nu0,
    Sumok is Sum-Sum0,

    NSum0 is (-1)*Sum0,

    nl(RF),write(RF,'%Number of binary relations:      '), write(RF,Nu),

```

```

nl(RF),write(RF,'%Number of unknown binary relations:      '), write(RF,Nu0),
nl(RF),write(RF,'%Number of reevaluated binary relations:      '), write(RF,Nuok),

nl(RF),write(RF,'%Uncertainty of unknown binary relations:      '), write(RF,NSum0),
nl(RF),write(RF,'%Uncertainty of reevaluated binary relations:      '), write(RF,Sumok),
close(RF),!.

fut(propose,To,[OldNu,Nu,RelOrder,SelStrategy],RL) :-
    format(atom(Output),'DW~c~a.pl',[92,To]),
    format(atom(File3),'Output file: ~a',[Output]),
    write(RL,File3),
    nl(RL),
    format(atom(Param),'Old rmax: ~a New rmax: ~a Order of pairs: ~a Selection strategy
~a',[OldNu,Nu,RelOrder,SelStrategy]),
    write(RL,Param),
    nl(RL),
    open(Output,write,RF,[buffering(none)]),
    propose(OldNu,Nu,RelOrder,SelStrategy,RL),

    final_state_max(RF),

    findall(N,rmax(Pers,N,Rep,Maxlist,E),Allrmaxlist),
    findall(N,rmax(Pers,N,Rep,Maxlist,[]),Newrmaxlist),
    findall(Rep,rmax(Pers,N,Rep,Maxlist,E),Allreplist),
    findall(Rep,rmax(Pers,N,Rep,Maxlist,[]),Newreplist),

    length(Allrmaxlist,Nall),
    length(Newrmaxlist,Nnew),

    sum_of_list(Allreplist,0,Sumall),
    sum_of_list(Newreplist,0,Sumnew),

    Repet is Sumall - Nall,
    nl(RF),write(RF,'%Number of maximal relations:      '), write(RF,Nall),
    nl(RF),write(RF,'%Number of suggested new maximal relations:      '), write(RF,Nnew),
    nl(RF),write(RF,'%Sum of maximal relations with repetition:      '), write(RF,Repet),
    nl(RF),write(RF,'%Sum of suggested new maximal relations with repetition:      '),
write(RF,Sumnew),
    close(RF),
    close(RL),!.

sum_of_list([],Sum0,Sum0) :- !.
sum_of_list([U|Other],Old,Sum0) :-
    New is Old+U,!,
    sum_of_list(Other,New,Sum0).

/***** reevaluation *****/

reevaluate(RL) :-
    rmax(_,ID,YN,Plist,[E]),
    evaluate(Plist,E),
    fail.
reevaluate(_).

evaluate(Plist,E) :-
    rmin(Pers,ID,y,_,Classes,[PropX,PropY],_,_,_,_),
    modify(Classes,[PropX,PropY],Plist,E),
    fail.
evaluate(_,_).

modify(Classes,[PropX,PropY],Plist,E) :-
    memberchk(PropX,Plist),
    memberchk(PropY,Plist),
    rmin(Pers,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,Unc,Vav,Forget),
    NFreq is Freq + 1,
    NVmin is min(Vmin,E),
    NVmax is max(Vmax,E),

```

```

    NUnc is NVmax-NVmin,
    averaging(NUnc,NVmax,NVmin,NVav),
    replace(
        rmin(Pers,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,UNC,Vav,Forget),
        rmin(Pers,ID,y,NFreq,Classes,[PropX,PropY],NVmin,NVmax,NUnc,NVav,Forget)),!.
modify(,_,_,_).

    averaging(NUnc,NVmax,NVmin,NVav) :-
        NUnc < 0,
        NVav is (-1)* 0.5,!.
    averaging(,_NVmax,NVmin,NVav) :-
        NVav is (NVmin+NVmax)/2,!.

/***** proposition *****/

propose(OldNu,Nu,RelOrder,SelStrategy,RL) :-
    new_solutions(OldNu,Nu,1,RelOrder,SelStrategy,RL).

new_solutions(,_0,_,_,_) :- !.
new_solutions(OldNu,Nu,N,RelOrder,SelStrategy,RL) :-
    findall(Pair,relexist(,_Pair),Pairlist),
    random_permutation(Pairlist,RandomPairlist),
    retractall(selr(,_,_,_)),
    nl(RL),nl(RL),
    write(RL,N),nl(RL),
    pair_selection(RelOrder,Pairlist,RandomPairlist,ActualPairlist),
    one_solution(ActualPairlist,SelStrategy,[],UMaxlist,[],Stratlist),
    sort(UMaxlist,Maxlist),
    nl(RL),write(RL,ActualPairlist),
    nl(RL),write(RL,Maxlist),
    nl(RL),write(RL,Stratlist),
    accept(OldNu,Nu,N,NN,Maxlist),
    NNu is Nu-1,!,
    new_solutions(OldNu,NNu,NN,RelOrder,SelStrategy,RL).

    pair_selection(normal_order,Pairlist,_,Pairlist) :- !.
    pair_selection(random_order,_,RandomPairlist,RandomPairlist) :- !.

accept(OldNu,Nu,N,N,Maxlist) :-
    rmax(Pers,ID,Rep,Maxlist,E),
    NRep is Rep+1,
    replace(rmax(Pers,ID,Rep,Maxlist,E),rmax(Pers,ID,NRep,Maxlist,E)),!.
accept(OldNu,Nu,N,NN,Maxlist) :-
    NN is N+1,
    New is OldNu+NN,
    assertz(rmax(dw,New,1,Maxlist,[])),!.
accept(,_,_,_).

one_solution([],_,Maxlist,Maxlist,Stratlist,Stratlist) :- !.
one_solution(,__,Maxlist,Maxlist,Stratlist,Stratlist) :-
    length(Maxlist,6),!.
one_solution([Pair|Otherpairs],SelStrategy,Oldmax,Maxlist,Oldstrat,Stratlist) :-
    strategy_selection(SelStrategy,Strategy,Aim),
    findall(ID,rmin(Pers,ID,_,_,Pair,_,_,_,_),IDList),
    retractall(selr(,_,_,_)),
    random_permutation(IDList,RIDList),
    class(RIDList,Strategy,Aim),
    classold(Oldmax,[],Cold),
    selects(Pair,Cold,Aim),
    retractall(selr(,_,_,_,[])),
    continues(Pair,Aim,Props),
    append(Props,Oldmax,Newmax),
    append(Oldstrat,[Strategy],Newstrat),!,
    one_solution(Otherpairs,SelStrategy,Newmax,Maxlist,Newstrat,Stratlist).

continues(Pair,Aim,Props) :-
    selr(ID,Pair,_,OV,PL),

```

```

    selected(Pair,Aim,OV),
    finds(Pair,Props),!.
continues(_,_,[ ]).

classold([],COLD,COLD) :- !.
classold([E|Other],Old,New) :-
    sub_atom(E,0,2,_,C),!,
    classold(Other,[C|Old],New).

class([],_,_) :- !.
class([ID|Other],Strategy,Aim) :-
    rmin(Pers,ID,y,Freq,Pair,[PropX,PropY], Vmin,Vmax,Unc,Vav,Forget),
    choosen_value(Strategy,Freq,Vmin,Vmax,Unc,Vav,Value),
    assertz(selr(ID,Pair,[PropX,PropY],Value,[ ])),!,
    class(Other,Strategy,Aim).

choosen_value(maxmin,Freq,Vmin,Vmax,Unc,Vav,Value) :-
    Value is Vmin,!.
choosen_value(maxmax,Freq,Vmin,Vmax,Unc,Vav,Value) :-
    Value is Vmax,!.
choosen_value(maxave,Freq,Vmin,Vmax,Unc,Vav,Value) :-
    Value is Vav,!.
choosen_value(minunc,Freq,Vmin,Vmax,Unc,Vav,Value) :-
    Value is Vmax-Vmin,!.
choosen_value(mintry,Freq,Vmin,Vmax,Unc,Vav,Value) :-
    Value is Freq,!.
choosen_value(unknown,0,Vmin,Vmax,Unc,Vav,Value) :-
    Value is -1,!.
choosen_value(unknown,Freq,Vmin,Vmax,Unc,Vav,Value) :-
    Value is Vav,!.

selects(Pair,COLD,Aim) :-
    selr(ID,Pair,[PropX,PropY],Value,[ ]),
    sub_atom(PropX,0,2,_,CPropX),
    sub_atom(PropY,0,2,_,CPropY),
    member(CPropX,COLD),
    member(CPropY,COLD),
    retractall(selr(ID,Pair,[PropX,PropY],Value,[ ])),!.

selects(Pair,COLD,Aim) :-
    selr(ID,Pair,[PropX,PropY],Value,[ ]),
    sub_atom(PropX,0,2,_,CPropX),
    sub_atom(PropY,0,2,_,CPropY),
    \+ member(CPropX,COLD),
    member(CPropY,COLD),

replace(selr(ID,Pair,[PropX,PropY],Value,[ ]),selr(ID,Pair,[PropX,PropY],Value,[PropX])),!.

selects(Pair,COLD,Aim) :-
    selr(ID,Pair,[PropX,PropY],Value,[ ]),
    sub_atom(PropX,0,2,_,CPropX),
    sub_atom(PropY,0,2,_,CPropY),
    member(CPropX,COLD),
    \+ member(CPropY,COLD),

replace(selr(ID,Pair,[PropX,PropY],Value,[ ]),selr(ID,Pair,[PropX,PropY],Value,[PropY])),!.

selects(Pair,COLD,Aim) :-
    selr(ID,Pair,[PropX,PropY],Value,[ ]),
    sub_atom(PropX,0,2,_,CPropX),
    sub_atom(PropY,0,2,_,CPropY),
    \+ member(CPropX,COLD),
    \+ member(CPropY,COLD),

replace(selr(ID,Pair,[PropX,PropY],Value,[ ]),selr(ID,Pair,[PropX,PropY],Value,[PropX,PropY])),!.

```

```

selects(,_,_).

selected(Pair,max,OV) :-
    selr(ID,Pair,PropList,V,PL),
    V >= OV,
    retractall(selr(ID,Pair,PropList,OV,PL)),
    asserta(selr(ID,Pair,PropList,V,PL)),
    fail.
selected(Pair,min,OV) :-
    selr(ID,Pair,PropList,V,PL),
    V <= OV,
    retractall(selr(ID,Pair,PropList,OV,PL)),
    asserta(selr(ID,Pair,PropList,V,PL)),
    fail.
selected(,_,_) :- !.

finds(Pair,Props) :-
    selr(,_,Pair,_,_,Props),!.

strategy_selection(random,Strategy,Aim) :-
    R is random(4), % actually minunc and mintry are not used
    selected_strategy(R,Strategy,Aim),!.
strategy_selection(SelStrategy,SelStrategy,Aim) :-
    selected_strategy(,_,SelStrategy,Aim),!.

selected_strategy(0,maxmin,max).
selected_strategy(1,maxmax,max).
selected_strategy(2,maxave,max).
selected_strategy(3,unknown,min).
selected_strategy(4,minunc,min).
selected_strategy(5,mintry,min).

relexist(1,[p1,p2]).
relexist(2,[p1,p3]).
relexist(3,[p1,p4]).
relexist(4,[p1,p5]).
relexist(5,[p1,p6]).
relexist(6,[p2,p3]).
relexist(7,[p2,p4]).
relexist(8,[p2,p5]).
relexist(9,[p2,p6]).
relexist(10,[p3,p4]).
relexist(11,[p3,p5]).
relexist(12,[p3,p6]).
relexist(13,[p4,p5]).
relexist(14,[p4,p6]).
relexist(15,[p5,p6]).

final_state_max(RF) :-
    save_max(RF).
final_state_max(_).

save_max(RF) :-
    rmax(Pers,N,YN,Maxlist,E),
    Term = rmax(Pers,N,YN,Maxlist,E),
    write_term(RF,Term,[],write(RF,' '),nl(RF)),
    fail.
save_max(_).

final_state_min(RF) :-
    save_min(RF).
final_state_min(_).

save_min(RF) :-
    rmin(Pers,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,Unc,Vav,Forget),
    Term = rmin(Pers,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,Unc,Vav,Forget),

```

```

    write_term(RF,Term,[]),write(RF,' '),nl(RF),
    fail.
save_min(_).

final_state_cons(RF) :-
    save_css(RF),
    save_cfl(RF),
    final_state_cons(_).

save_css(RF) :-
    rmin(css,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,Unc,Vav,Forget),
    Term = rmin(css,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,Unc,Vav,Forget),
    write_term(RF,Term,[]),write(RF,' '),nl(RF),
    fail.
save_css(_).
save_cfl(RF) :-
    rmin(cfl,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,Unc,Vav,Forget),
    Term = rmin(cfl,ID,y,Freq,Classes,[PropX,PropY],Vmin,Vmax,Unc,Vav,Forget),
    write_term(RF,Term,[]),write(RF,' '),nl(RF),
    fail.
save_cfl(_).

replace(Term,NewTerm) :-
    retractall(Term),
    assertz(NewTerm),!.

```

A3: Details about the simplified illustrative example for learning from the RPS

This Appendix contains some additional details to Section 5.1.

Property equivalence classes:

p1, p2 and p3.

Compatibility relations amongst these property classes:

relexist(1,[p1,p2]).
relexist(2,[p1,p3]).
relexist(3,[p2,p3]).

Properties in the property classes:

p1_1, p1_2, p1_3 in equivalence class p1;
p2_1, p2_2, p2_3, p2_4 in equivalence class p2;
p3_1, p3_2 in equivalence class p3.

Definition of minimal (binary) relations:

rmin(XY,ID,YN,Freq,Classes,Properties,Vmin,Vmax,Unc,Vav,Forget).

where

XY = the identifier of evaluator;
ID = an identifying integer;
YN = by default y if it is considered during the evaluation, otherwise it is n;
Freq = number of evaluations (by default 0 at the beginning, and will be incremented during the evaluations);
Vmin = minimal value (m) during the run (initial lack of knowledge is considered by Vmin = 1);
Vmax = maximal value (M) during the run (initial lack of knowledge considered by Vmax = 0);
Unc = uncertainty, $Unc = Vmax - Vmin$;
Vav = average value, $Vav = sign(Unc) * (Vmax + Vmin) / 2$;
Forget = number of forgetting steps, by default 0 (temporarily we don't use this).

Binary relations between the properties of compatible property classes, initialized for expert e1 by 'lack of knowledge':

rmin(e1,1,y,0,expert e1,[p1,p2],[p1_1,p2_1],1,0,-1,-0.5,0).
rmin(e1,2,y,0,[p1,p2],[p1_1,p2_2],1,0,-1,-0.5,0).
rmin(e1,3,y,0,[p1,p2],[p1_1,p2_3],1,0,-1,-0.5,0).
rmin(e1,4,y,0,[p1,p2],[p1_1,p2_4],1,0,-1,-0.5,0).
rmin(e1,5,y,0,[p1,p2],[p1_2,p2_1],1,0,-1,-0.5,0).
rmin(e1,6,y,0,[p1,p2],[p1_2,p2_2],1,0,-1,-0.5,0).
rmin(e1,7,y,0,[p1,p2],[p1_2,p2_3],1,0,-1,-0.5,0).

```

rmin(e1,8,y,0,[p1,p2],[p1_2,p2_4],1,0,-1,-0.5,0).
rmin(e1,9,y,0,[p1,p2],[p1_3,p2_1],1,0,-1,-0.5,0).
rmin(e1,10,y,0,[p1,p2],[p1_3,p2_2],1,0,-1,-0.5,0).
rmin(e1,11,y,0,[p1,p2],[p1_3,p2_3],1,0,-1,-0.5,0).
rmin(e1,12,y,0,[p1,p2],[p1_3,p2_4],1,0,-1,-0.5,0).
rmin(e1,13,y,0,[p1,p3],[p1_1,p3_1],1,0,-1,-0.5,0).
rmin(e1,14,y,0,[p1,p3],[p1_1,p3_2],1,0,-1,-0.5,0).
rmin(e1,15,y,0,[p1,p3],[p1_2,p3_1],1,0,-1,-0.5,0).
rmin(e1,16,y,0,[p1,p3],[p1_2,p3_2],1,0,-1,-0.5,0).
rmin(e1,17,y,0,[p1,p3],[p1_3,p3_1],1,0,-1,-0.5,0).
rmin(e1,18,y,0,[p1,p3],[p1_3,p3_2],1,0,-1,-0.5,0).
rmin(e1,19,y,0,[p2,p3],[p2_1,p3_1],1,0,-1,-0.5,0).
rmin(e1,20,y,0,[p2,p3],[p2_1,p3_2],1,0,-1,-0.5,0).
rmin(e1,21,y,0,[p2,p3],[p2_2,p3_1],1,0,-1,-0.5,0).
rmin(e1,22,y,0,[p2,p3],[p2_2,p3_2],1,0,-1,-0.5,0).
rmin(e1,23,y,0,[p2,p3],[p2_3,p3_1],1,0,-1,-0.5,0).
rmin(e1,24,y,0,[p2,p3],[p2_3,p3_2],1,0,-1,-0.5,0).
rmin(e1,25,y,0,[p2,p3],[p2_4,p3_1],1,0,-1,-0.5,0).
rmin(e1,26,y,0,[p2,p3],[p2_4,p3_2],1,0,-1,-0.5,0).

```

Definition of maximal relations (complete combinations):

`rmax(XY,ID,Rep,Plist,[Eval],[]).`

where

XY = 0, if the propositions form all binary knowledge bases (i.e. from all evaluators) are considered in a common file (we apply this case in this example), otherwise XY signs the individual identifier of the evaluator, while the evaluators use their own `rmax()` databases.
 XY = ori means that the database contains the hypothetical “true” (i.e. previously known) evaluation of the given combination;

ID = identifying integer of the combination;

Rep = the number of repeated occurrences of the given complete combination. Inial Rep = 0 means that it has not already proposed;

Plist = the list of properties, representing the given combination (one and only one property from each equivalence class, ordered according to the equivalence classes);

Eval = evaluation of the given complete combination.

Maximal relations (complete combinations), synthesized from the binary relations:

```

rmax(ori,1,0,[p1_1,p2_1,p3_1],[0.359],[]).
rmax(ori,2,0,[p1_1,p2_1,p3_2],[0.362],[]).
rmax(ori,3,0,[p1_1,p2_2,p3_1],[0.564],[]).
rmax(ori,4,0,[p1_1,p2_2,p3_2],[0.156],[]).
rmax(ori,5,0,[p1_1,p2_3,p3_1],[0.716],[]).
rmax(ori,6,0,[p1_1,p2_3,p3_2],[0.394],[]).
rmax(ori,7,0,[p1_1,p2_4,p3_1],[0.200],[]).
rmax(ori,8,0,[p1_1,p2_4,p3_2],[0.436],[]).
rmax(ori,9,0,[p1_2,p2_1,p3_1],[0.218],[]).
rmax(ori,10,0,[p1_2,p2_1,p3_2],[0.855],[]).

```

```

rmax(ori,11,0,[p1_2,p2_2,p3_1],[0.349],[ ]).
rmax(ori,12,0,[p1_2,p2_2,p3_2],[0.651],[ ]).
rmax(ori,13,0,[p1_2,p2_3,p3_1],[0.418],[ ]).
rmax(ori,14,0,[p1_2,p2_3,p3_2],[0.127],[ ]).
rmax(ori,15,0,[p1_2,p2_4,p3_1],[0.840],[ ]).
rmax(ori,16,0,[p1_2,p2_4,p3_2],[0.257],[ ]).
rmax(ori,17,0,[p1_3,p2_1,p3_1],[0.671],[ ]).
rmax(ori,18,0,[p1_3,p2_1,p3_2],[0.217],[ ]).
rmax(ori,19,0,[p1_3,p2_2,p3_1],[0.376],[ ]).
rmax(ori,20,0,[p1_3,p2_2,p3_2],[0.451],[ ]).
rmax(ori,21,0,[p1_3,p2_3,p3_1],[0.243],[ ]).
rmax(ori,22,0,[p1_3,p2_3,p3_2],[0.617],[ ]).
rmax(ori,23,0,[p1_3,p2_4,p3_1],[0.252],[ ]).
rmax(ori,24,0,[p1_3,p2_4,p3_2],[0.376],[ ]).

```

while "ori" designates that these randomly selected evaluations are saved and used in the various evaluating case studies.

Rules, defining the erroneous and subjective evaluations of e1, e2,..., e8 simulated experts:

```

expert(e1,E0,E) if L is E0-E0/3,U is E0+(1-E0)/3,random(L,U,E).
expert(e2,E0,E) if L is E0-E0/4,U is E0+(1-E0)/4,random(L,U,E).
expert(e3,E0,E) if L is E0-E0/5,U is E0+(1-E0)/5,random(L,U,E).
expert(e4,E0,E) if L is E0-E0/2.5,U is E0+(1-E0)/2.5,random(L,U,E).
expert(e5,E0,E) if L is E0-E0/3.5,U is E0+(1-E0)/3.5,random(L,U,E).
expert(e6,E0,E) if L is E0-E0/4.5,U is E0+(1-E0)/4.5,random(L,U,E).
expert(e7,E0,E) if L is E0-E0/2,U is E0+(1-E0)/2,random(L,U,E).
expert(e8,E0,E) if L is E0-E0/5.5,U is E0+(1-E0)/5.5,random(L,U,E).

```

where

E0 = the previously generated hypothetical "true" value of the given combination;
L = the lower bound of the random range for the evaluators around the E0 value;
U = the upper bound of the random range for the evaluators around the E0 value;
E = the simulated, possibly erroneous and subjective evaluation of the experts.

Illustrations of the case study for 3 simulated experts

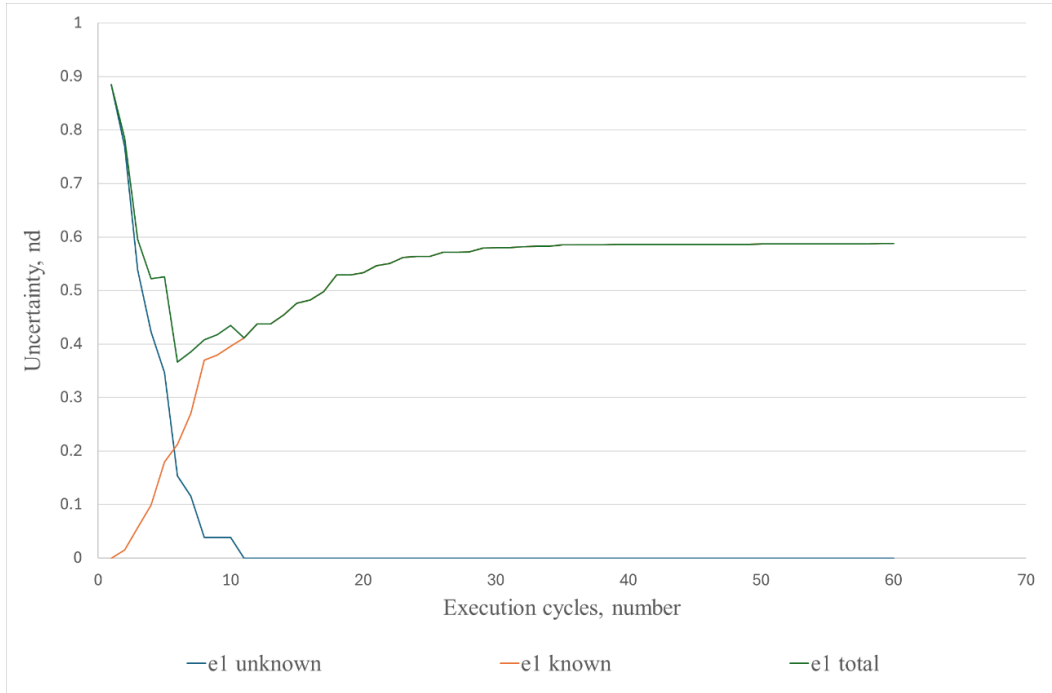


Figure 19 Various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators

Figure 19 summarizes the various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators. Similarly to Figure 18 in Chapter 5.9 of this dissertation, the uncertainty of “lack of knowledge” decreases, and the uncertainty of knowledge increases, while the total uncertainty first decreases, and after a minimum value, increases toward a limit value.

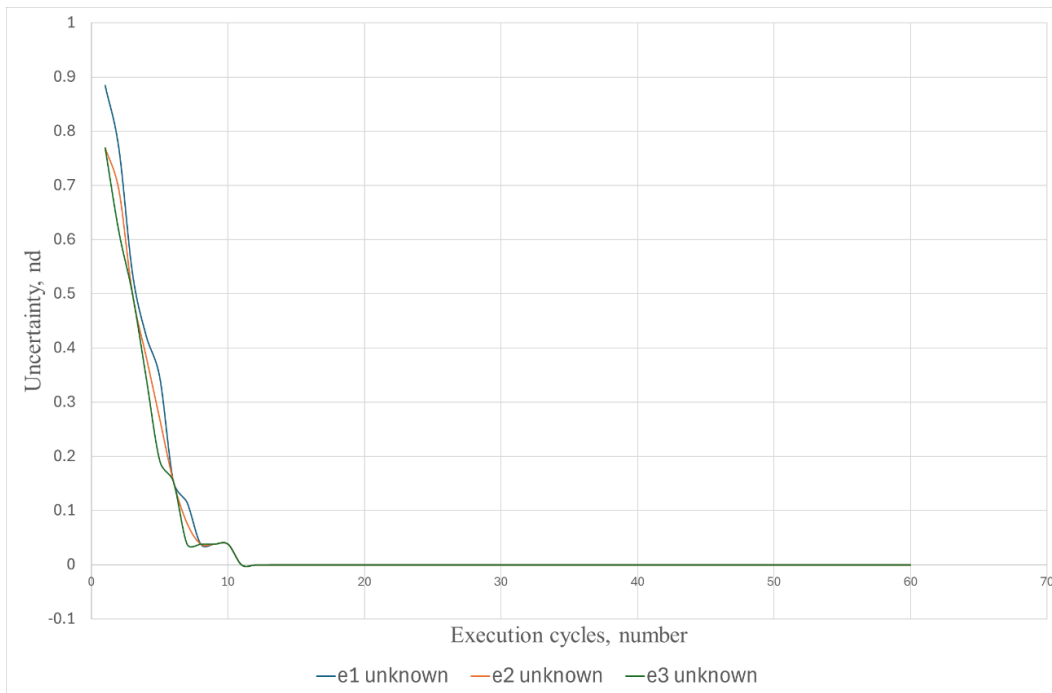


Figure 20 Uncertainty of “lack of knowledge” from 3 evaluators

The uncertainty of “lack of knowledge” for 3 evaluators are illustrated in Figure 20. The quick decrease to zero is a consequence of the simple example behind the study.

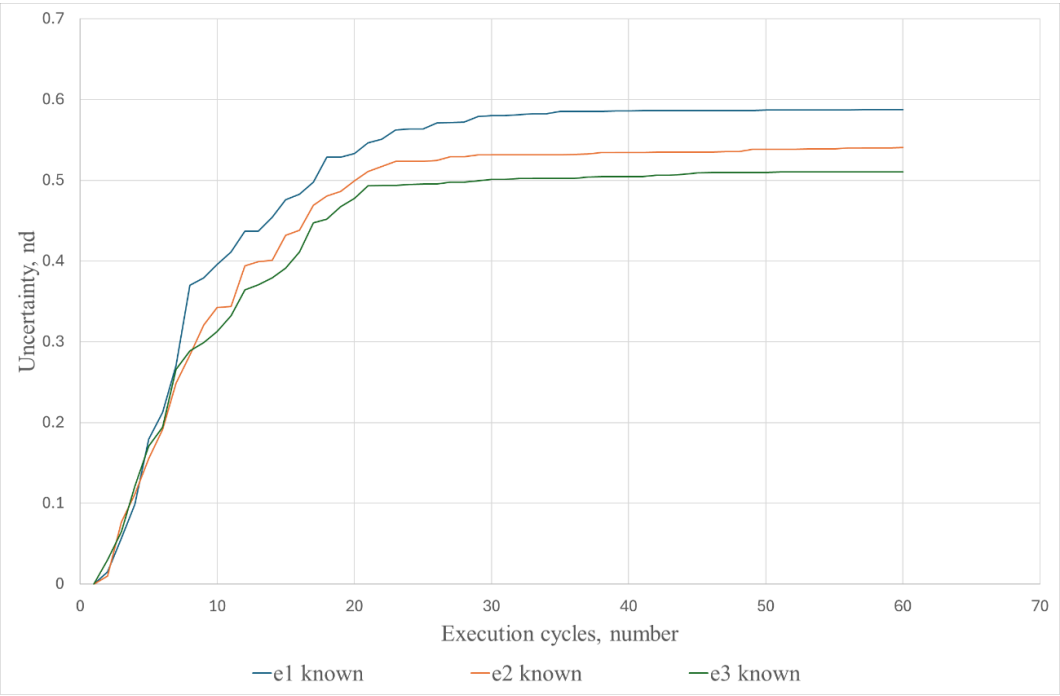


Figure 21 Uncertainty of available knowledge for 3 evaluators

Figure 21 shows the uncertainty of available knowledge for 3 evaluators. Each case goes toward a limit value, while their difference is the consequence of differently “erroneous” and “subjective” simulated evaluations.

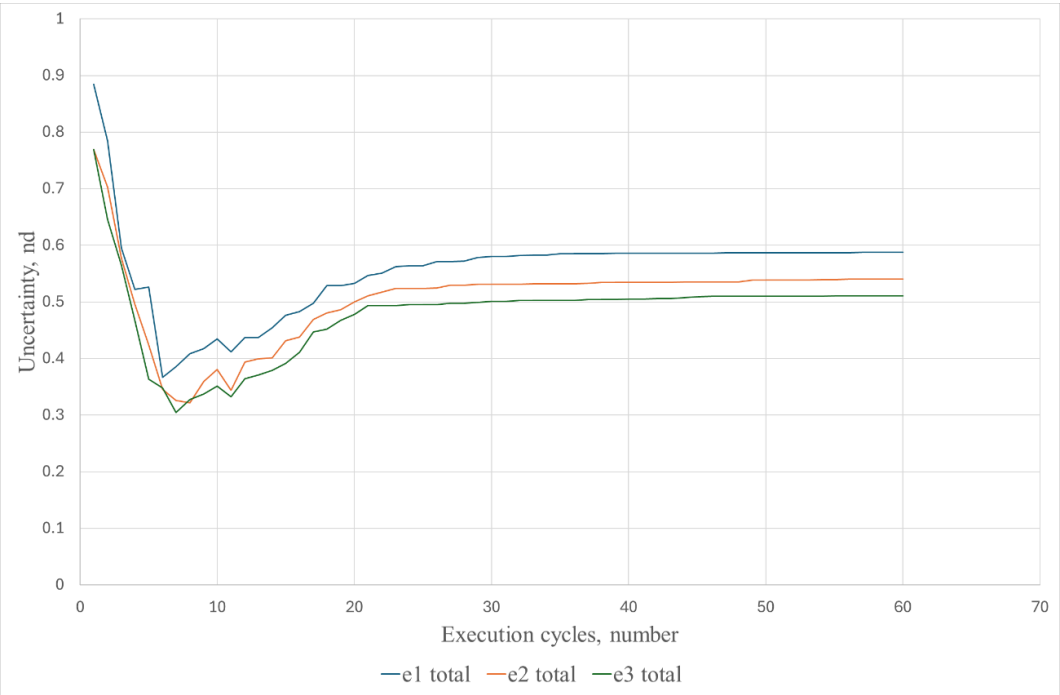


Figure 22 Total uncertainty for 3 evaluators

The total uncertainty, according to Figure 22, expresses the same characteristics.

Illustrations of the case study for 3 simulated experts with cyclically repeated consent



Figure 23 Various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators, utilizing stepwise consent

Figure 23 summarizes the various kinds of uncertainties of knowledge for evaluator e1, in case of 3 evaluators, utilizing stepwise consent. As it can be seen, consensus decreases uncertainty and the limit value is also less than in the no-consensus case (compared with Figure 6). Figures 24-26 illustrate the effect of stepwise consensus for the uncertainty of ‘lack of knowledge’, available knowledge and their sum, respectively.

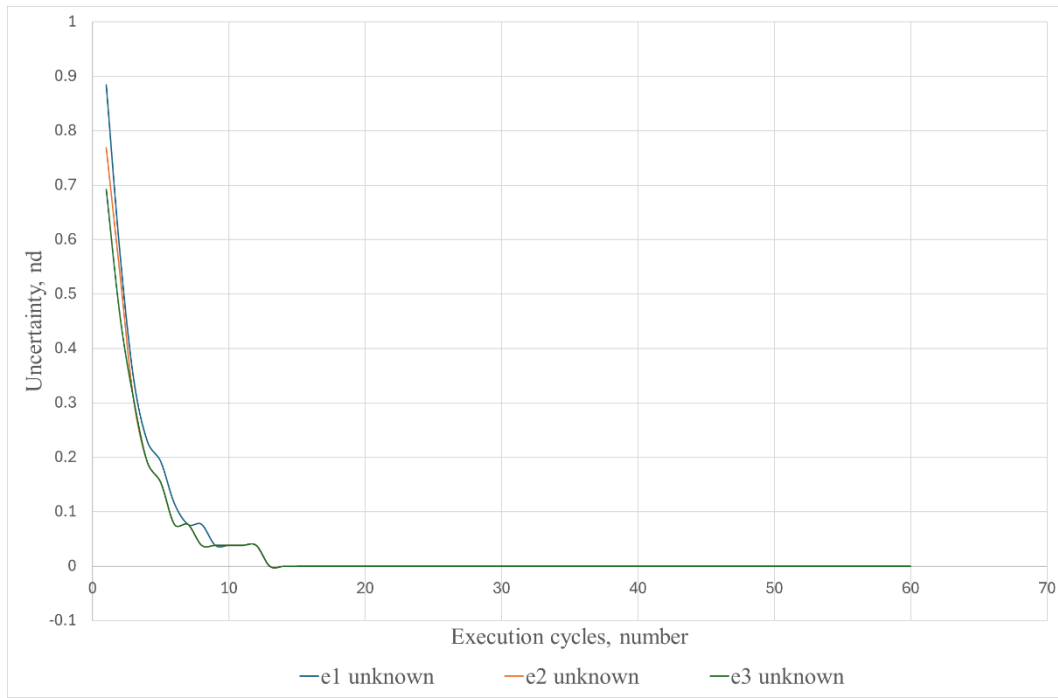


Figure 24 Uncertainty of lack of knowledge, in case of 3 evaluators, utilizing stepwise consent

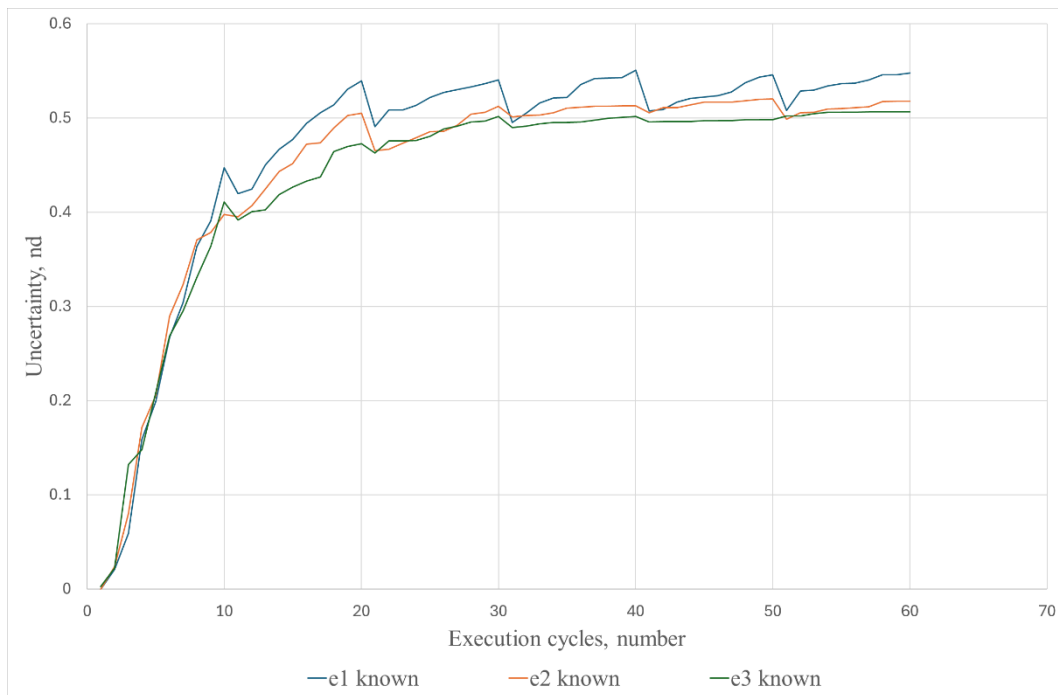


Figure 25 Uncertainty of available knowledge, in case of 3 evaluators, utilizing stepwise consent

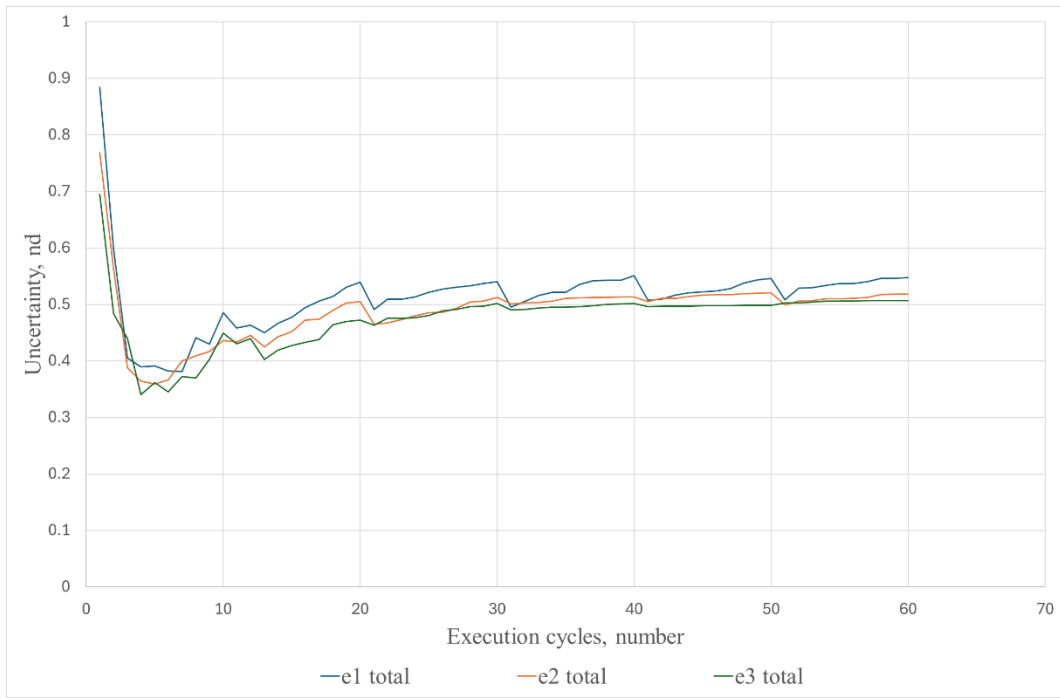


Figure 26 Sum of uncertainty, in case of 3 evaluators, utilizing stepwise consent

Illustrations of the case study for 8 simulated experts

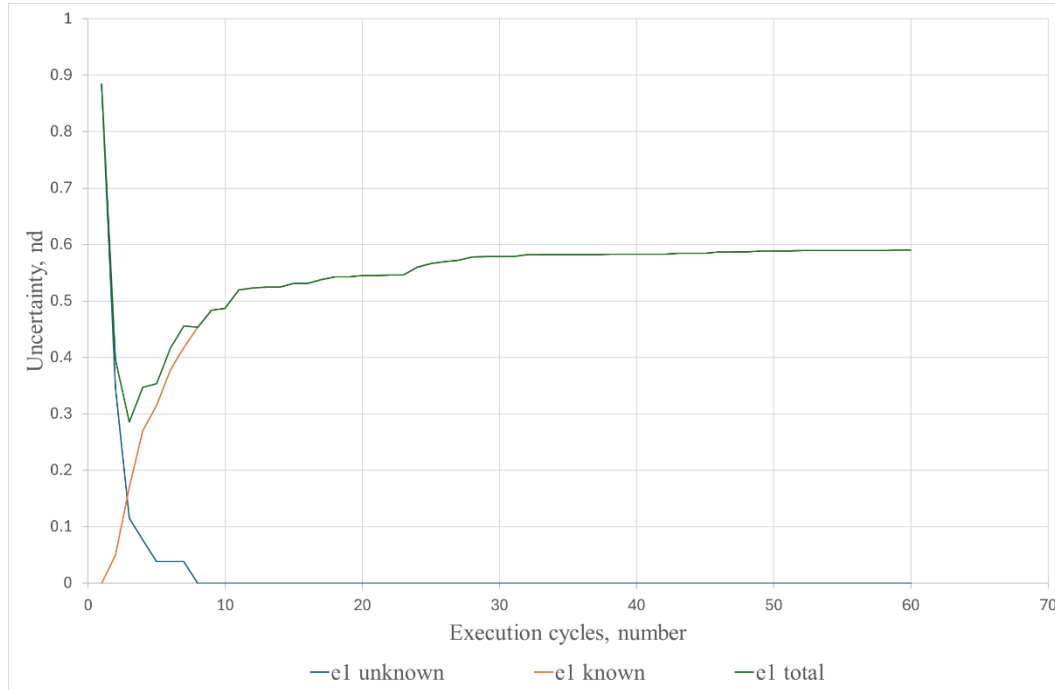


Figure 27 Various kinds of uncertainties of knowledge for evaluator *e1*, in case of 8 evaluators

The simulations with 8 evaluators do not show any major differences. According to Figure 27, the asymptotic behavior of evaluator *e1* is almost the same as in case of 3 evaluators (it is to be mentioned, that they are not identical, because more evaluators cause more changes in the randomly selected sequence of proposed complete combinations that influences the evolution of the individual binary knowledge bases, too). Figures 28-30 illustrate that more evaluators represent an increasing range in the development of their individual binary knowledge bases.

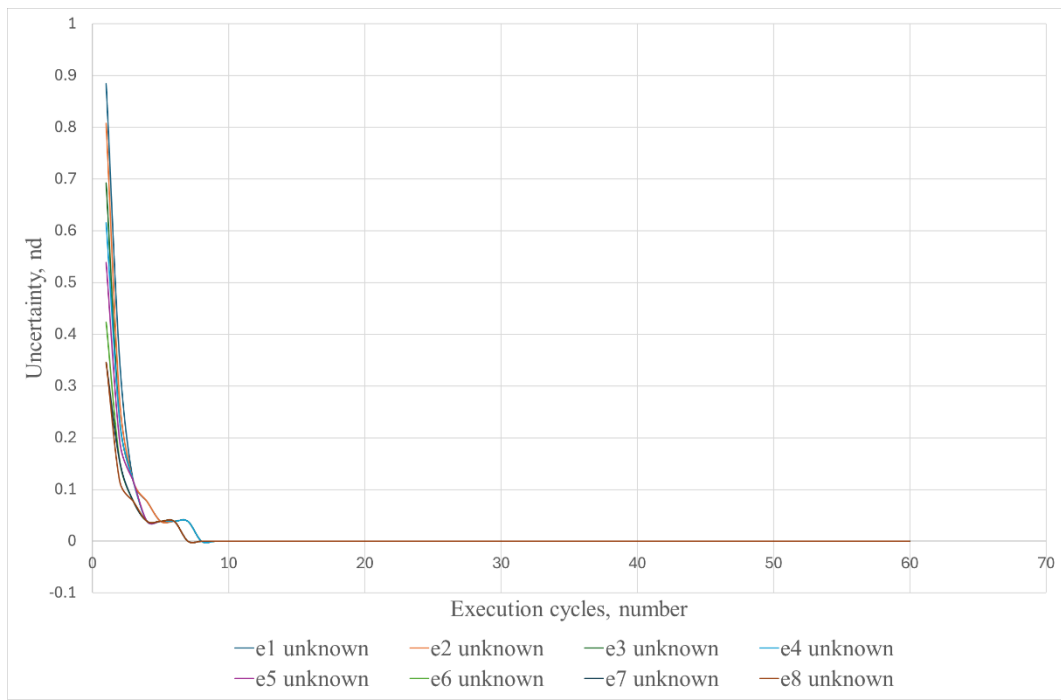


Figure 28 Uncertainty of lack of knowledge from 8 evaluators

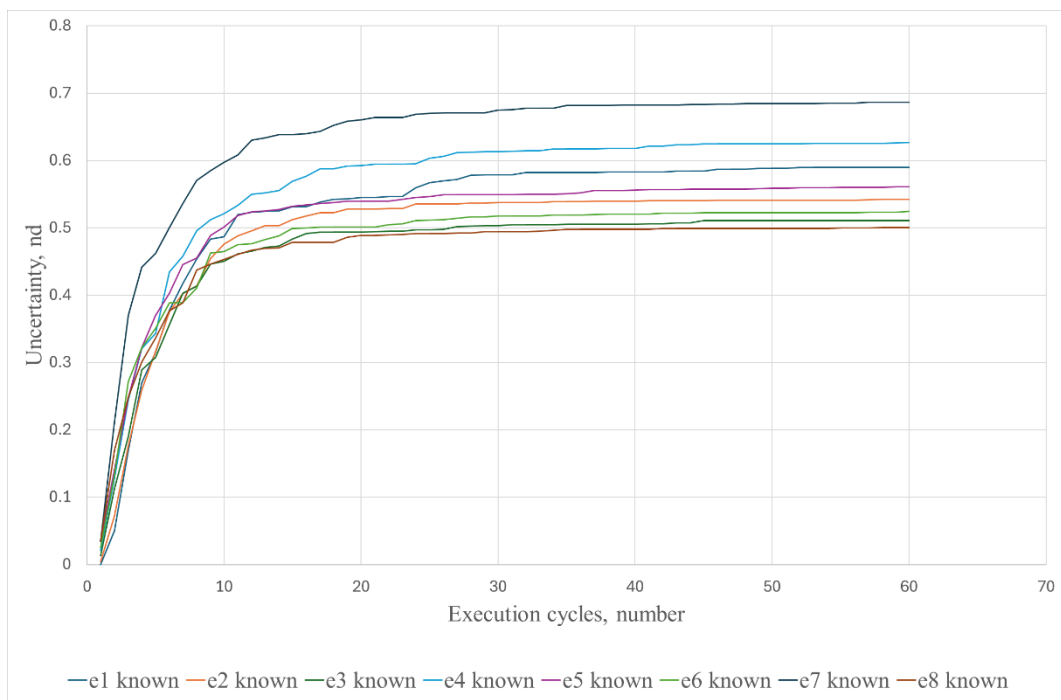


Figure 29 Uncertainty of available knowledge for 8 evaluators

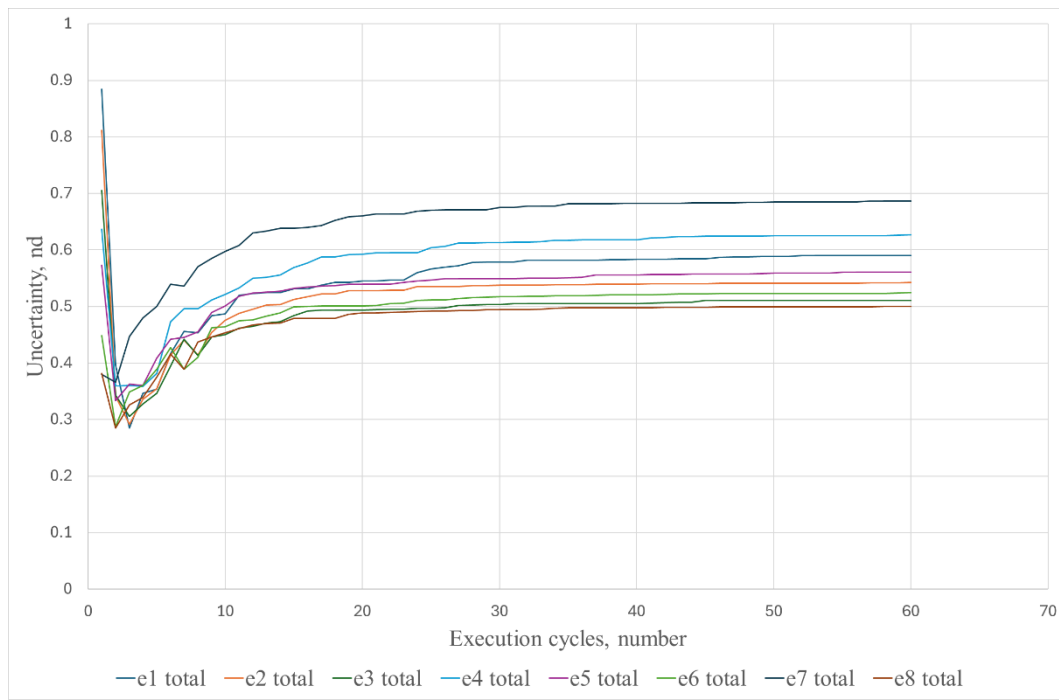


Figure 30 Total uncertainty for 8 evaluators

Illustrations of the case study for 8 simulated experts with cyclically repeated consent



Figure 31 Various kinds of uncertainties of knowledge for evaluator e1, in case of 8 evaluators, utilizing stepwise consent

Figure 31 shows the various kinds of uncertainties of knowledge for evaluator e1, in case of 8 evaluators, utilizing stepwise consent. As it can be seen, consensus decreases uncertainty similarly, and the limit value is also less than in the no-consensus case (compared with Figure 27). Figures 32-34 illustrate the effect of stepwise consensus for the uncertainty of ‘lack of knowledge’, available knowledge and their sum, respectively.

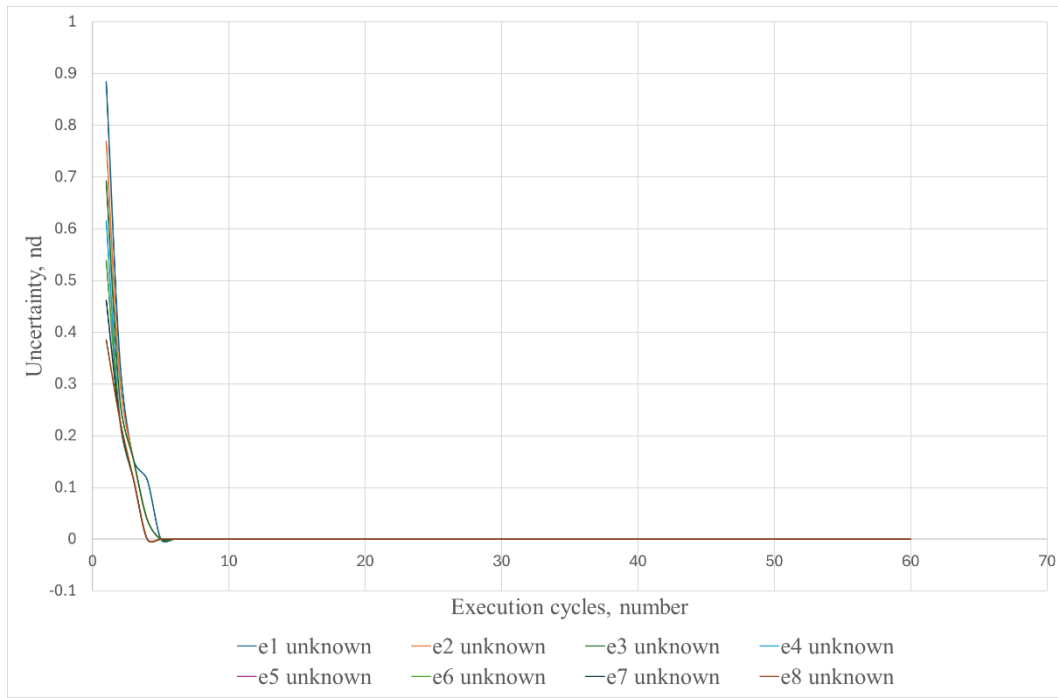


Figure 32 Uncertainty of lack of knowledge, in case of 8 evaluators, utilizing stepwise consent

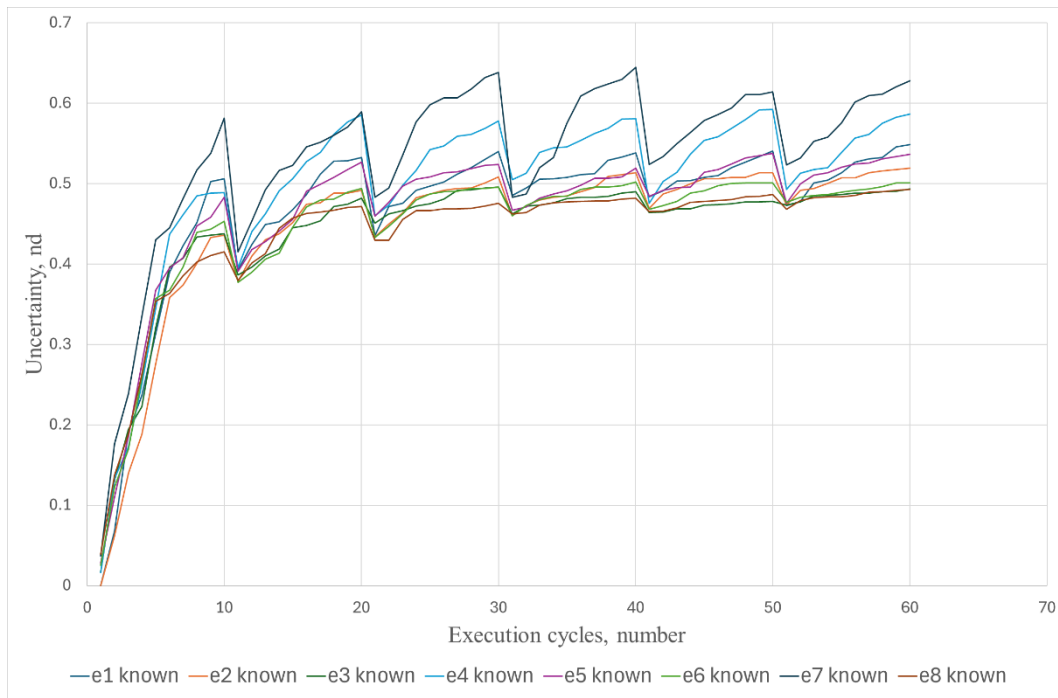


Figure 33 Uncertainty of available knowledge, in case of 8 evaluators, utilizing stepwise consent

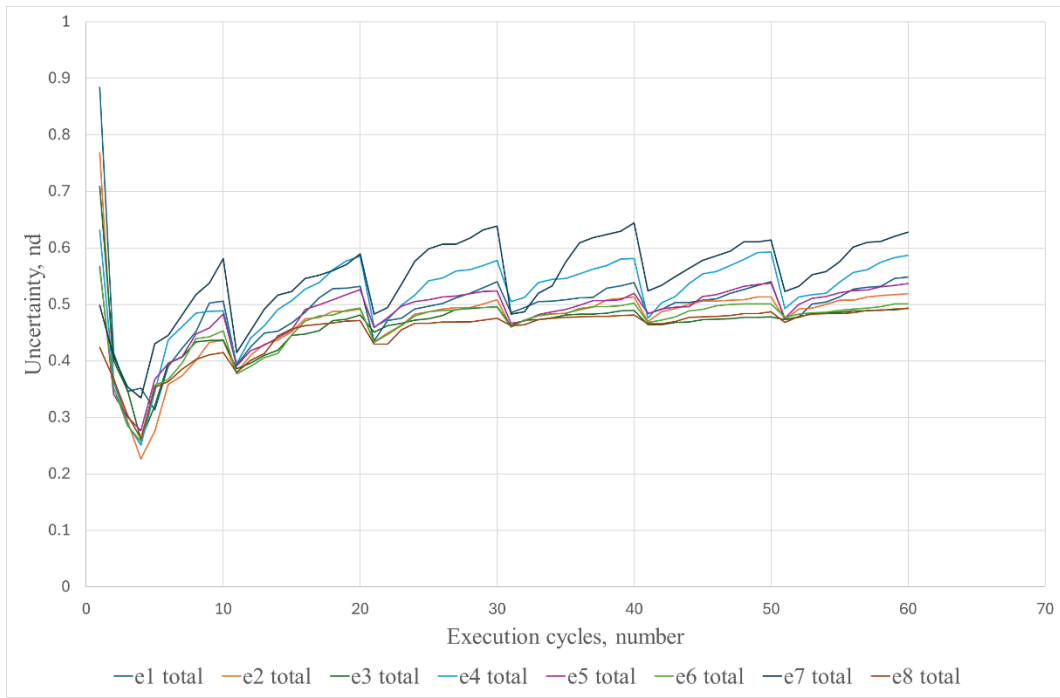


Figure 34 Total uncertainty for 8 evaluators, utilizing stepwise consent

A4: Related publications

Scientific papers in the topic of this dissertation

WEBER, D., AND VARGA, M. 2025a. A Reevaluable Property Lattice-Based Knowledge Representation for Proposing and Assessing Computational Tools in Manufacturing. In: Machine Learning and Knowledge Extraction: 7 (4), 161-196 (2025).

WEBER, D. (2022). The impact of artificial intelligence on business organizations. In: Berke, S., Szabó, K., Pató, G., Szücs, B. (eds.). *Leadership and Management Theory in Practice* (2022), 41-51.

WEBER, D. (2021). Artificial Intelligence and the UNs Sustainable Development Goal #9. In: Hammerl, L., Kremer, O., Weber, D. *Fulfilling the Sustainable Development Goals (SDGs) of the United Nations through hydrogen innovation, economic growth, and technological breakthrough* (2021), 32-51, GRIN Publishing, ISBN: 9783346615411.

TON, A., HAMMERL, L., WEBER, D., KREMER, O., SZABO-SZENTGROTI, G. (2022). Why leaders are important for cross-functional teams: Moderating role of supportive leadership on knowledge hiding. *Problems and Perspectives in Management*: 20 (3), 178-191 (2022).

HAMMERL, L., WEBER, D., TON, A. (2021). Kaizen in Automotive Innovation: How the Hungarian Automotive Clusters Can Profit from the Adoption of Kaizen Principles – A Literature Review of the Central European Automotive Industry. In: *International Journal of Applied Research in Business and Management*: 2 (3), 23-38 (2021).

Published data and code in the topic of the dissertation

WEBER, D., VARGA, M. (2025b). RPS_algorithm.
https://github.com/monikavarga/RPS_algorithm

Conference proceedings

HAMMERL, L., KREMER, O., WEBER, D. (2022). Innovation in the Republic of Turkey: Strengths and Barriers of the Domestic Expansion of the Automotive Industry. In: Bakan, I., Kocagöz, Ü. (eds.) *21st International Business Congress Proceedings Book* (2022), 1011-1016.

WEBER, D., HAMMERL, L. (2021). Artificial Intelligence in agriculture - Why farmers will soon become AI experts. In: CASEE CONFERENCE 2021 Book of Abstracts : “CASEE universities as laboratories for new paradigms in life sciences and related disciplines” (2021), 18-29.

WEBER, D., HAMMERL, L., TON, A., (2021). Business resilience through AI expertise – an opportunity for rural economies? In: Pop, G., Bíró, B., Csata, A., György, O., Kassay, J., Koroseczné, P., Madaras, S., Pál, L., Péter, K., Szőcs, A., Tánczos, L., Telegdy, B., (eds.). *Challenges in the Carpathian Basin : global challenges - local answers : interdependencies or globalisation?: 15th International Conference on Economics and Business (2021)*, 192-204.

KREMER, O., HAMMERL, L., WEBER, D. (2022). Implementation of cyber-physical systems: Health innovation and the adaption of hydrogen technologies in the work processes of hospitals. In: Bakan, I., Kocagöz, Ü. (eds.) *21st International Business Congress Proceedings Book (2022)*, 1011-1016.

KREMER, O., HAMMERL, L., WEBER, D. (2022). Possibilities of modern health innovation: Reducing the shortage of skilled workers in the health care sector through migration. In: *V. BBS International Sustainability Student Conference (2022)*.

TON, A., WEBER, D., HAMMERL, L. (2021). How coopetition of cross-functional teams becomes important in times of COVID-19 in carpathian basin: A Grounded Theory. In: Pop, G., Bíró, B., Csata, A., György, O., Kassay, J., Koroseczné, P., Madaras, S., Pál, L., Péter, K., Szőcs, A., Tánczos, L., Telegdy, B., (eds.). *Challenges in the Carpathian Basin : global challenges - local answers : interdependencies or globalisation?: 15th International Conference on Economics and Business (2021)*, 810-834.