EPFL

Thèse n° 7945

Automated Design: A Journey Across Modelling, Optimization, and Education

Présentée le 25 juin 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire de conception mécanique appliquée
Programme doctoral en énergie

pour l'obtention du grade de Docteur ès Sciences

par

Cyril PICARD

Acceptée sur proposition du jury

Prof. F. Maréchal, président du jury
Prof. J. A. Schiffmann, directeur de thèse
Prof. C. A. Coello Coello, rapporteur
Prof. F. Ahmed, rapporteur
Prof. M. Bierlaire, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

Un Grand Opéra en quatre actes et prélude

En mémoire de mon ami et collègue Antonio...

Acknowledgements

When I started my PhD in June 2016, little did I know where the journey would bring me, and what a great human experience it would be. Numerous people have participated in making it inspiring, enjoyable, dynamic and fun. I would like to take the opportunity to acknowledge them here.

I wish to express my sincere gratitude to my thesis director Prof. Jürg Schiffmann. Jürg, you have supported me personally through complicated meetings, and supported my work with closed eyes—*were they really closed?*—when I embarked you in a crazy project with your students. I'm glad you did: it ended up being an important contribution in my thesis.

I also wish to express my warmest gratitude to my co-authors: Dr. Violette Mounier, Dr. Cécile Hardebolle, and Dr. Roland Tormey. Our collaborations have been a real enriching experience that was a great breath of fresh air compared to the traditionally more solitary work of a PhD. Violette, I think we hold the record for the most productive research ever done at LAMD: from nothing to best-paper award in two months. Cécile and Roland, for me, it was a great adventure outside paved roads into the fascinating world of education and psychometrically validated tools. Cécile, thank you for all your help into setting up the projects with the students and for co-driving this important research with me.

I am glad to have joined the LAMD, not only for the research, but also for the people. Certainly, Dr. Karim Shalash and Dr. Patrick Wagner opened the door first by being kind, but demanding supervisors of my semester project at the lab during my Master. Karim, we ended up almost sharing desks during a large part of our PhDs, a time of intense scientific debates. You have pushed me to search for the root cause of every problem in general, and in particular, you pushed me out of my comfort zone into the "HAL 9001" project. I don't know if I have found the root cause, but I still enjoy searching for it. As for this project, Chapter 5 is for you! I also had the pleasure to have an incredible companion from our first years at EPFL as young students, to our years as PhD students at LAMD. Thank you, Dr. Eliott Guenat! I feel honored to have been able to share this epic journey with such a complete scientist and person as you are. I had the honor to share my time at the lab with Antonio Lopez. Thank you for your endless support and positive energy. You taught us to always see the glass half full, and I will try to follow your teaching. For their friendliness, efficiency, and support, I thank Cécile Taverney from the doctoral program in energy, and Dr. Julie Lenoble, the accountant, counselor, event manager, psychologist, real estate agent, and secretary of LAMD.

I would also like to thank all the coffee break aficionados. However we triggered them—“café ?”, “coffee?”, “coffee guys?”, “*covfefe*?”[1]—the coffee breaks were inspiring, intriguing, dangerous, scary, factual, heated, light-minded, political, depressive, enriching and most importantly fun. So thank you to all my past and present colleagues at LAMD: Adeel, Ansgar, Antonio, Christoph, Christophe D., Christophe P., Ceyhun, Eliott, Eric, Julie, Jürg, Karim, Kévin, Kossi, Lili, Luis, Markus, Nathaniel, Nicolas, Patrick, Philipp, Phillip, Sajjad, Soheyl, Suresh, Thierry, Tomohiro, Victoria, Violette, and Wanhui.

A substantial part of my work has revolved around students, up to the point that engineering education research became an integral part of this thesis. Here, I want to thank the many students from the courses of “Systèmes Mécaniques” and of “Applied Mechanical Design”, as well as the students I supervised for their project. You have made me sweat, you have made me smile, but you have shown me that education can have a direct impact on our world, one step at a time. I would like in particular to thank Guillaume Spaeth and Eugène Lemaitre, who have, through their Master thesis, contributed to my research project. I would also like to thank all 205 students who accepted to participate in our study: I owe you Chapter 5. My interest in education has also been fueled by the lectures by Siara Isaac and Cécile Hardebolle from the “Centre d’Appui pour l’Enseignement”.

To maintain my mental well-being, I could count on many amazing friends. There are my “EPFL friends” that went on to do a PhD—including (Dr.) Frederike Dümbgen, (Dr.) Baptiste Ulrich, Dr. Guillaume Jeanmonod, and (Dr.) Sarah van Rooij—among whom we shared tips and our frustrations and successes. It was also good to have friends outside the science bubble. Here, I want to thank in particular: Anouk, Aurélie, Clément, and Simon. Despite the distance, thank you for the dinners, vacations, and discussions.

I thank my parents Katharina and Didier; my most faithful readers and advisors. I also thank my sister Melanie, her husband Cyril—it is not a typo—and their son Martin. Our little getaway to Santa Monica beach was the refreshment I needed to clear my mind to write this thesis. I thank my girlfriend’s mother Verena for her indirect and direct contribution—thank you for the references on intelligence.

Jasmine, what an exciting journey it has been to share all these moments with you, in our cosy apartment “au jardin”, or on the go. You were my greatest support through the years and the many challenges, thank you. I am impatient for the next adventures: finishing your thesis, raising our son, and much more.

This work was supported by Innosuisse through a project with Johnson Electric International AG. I thank their team in Murten for our fruitful collaboration. And finally, I thank Prof. Michel Bierlaire, Prof. Carlos A. Coello Coello, Prof. Faez Ahmed, and Prof. François Maréchal for having taken part in my thesis jury, for your helpful feedback, and your encouragements.

Bern, April 19, 2021
C.P.

---

[1] This is our interpretation of the famous tweet: https://en.wikipedia.org/wiki/Covfefe

Abstract

Machine intelligence greatly impacts almost all domains of our societies. It is profoundly changing the field of mechanical engineering with new technical possibilities and processes. The education of future engineers also needs to adapt in terms of techniques and even skills.

Using the design of electro-mechanical actuators as a common thread, this work explores the many-facets of automated design: modeling, optimization, and education, and looks for the prerequisites essential to its successful application.

The journey starts by building a modular and integrated model. It focuses on the prediction of system-level specifications that yield high added-value for decision-makers and shorten the path from the model to the final product. Combined with multiobjective evolutionary algorithms (MOEAs) and visualization tools, the model forms an automated design tool that helps engineers and decision-makers to rapidly get important insights into their design task. Its potential and benefits are validated through two specific applications. The results, however, also highlight a gap between the reported performance of optimizers on common benchmark problems and the actual performance on these problems.

To further develop optimizers, appropriate and realistic benchmark problems are needed. A subset of the integrated design model is used to formulate a new test suite called MODAct, composed of 20 constrained multiobjective optimization problems (CMOPs) with variable levels of complexity. In addition, numerical approaches to evaluate the constraint landscape of CMOPs are introduced and applied to identify the differences in features of MODAct against 45 benchmark problems from literature. Further, the convergence performance of three algorithms on the same problems highlights the key role of constraints and, specifically, the number of simultaneously violated constraints in MODAct problems.

In a next step, existing constraint handling strategies suitable for MOEAs along with a newly proposed technique for many-constraint problems are evaluated. Their parameters are tuned for different problems. The performance of the various configurations further highlights the difference between MODAct and other benchmark problems and show the highly competitive results of the proposed constraint handling technique on realistic design problems.

As the technical limits are removed, the impact of automated design on the work of future engineers should be considered. On the one hand, the development of professional skills by students working on team project in different settings has been evaluated thanks to 205 students from three classes. Explicitly addressing these skills within the project seems key to support stronger and broader learning, suggesting changes that do not require a full curriculum redesign. On the other hand, nine groups (33 students) have been asked to design an actuator using a conventional approach followed by an automated design approach. The actuators suggested by students using the automated tool outperform the designs obtained through the traditional approach. Six groups even suggest solutions cheaper, three of which are also smaller, than the product of experienced industry engineers. Students proved thus capable of leveraging the tool within a short time. The analysis of their mistakes suggests possible improvements for future tools. As these students leave university, they carry the hope to see such methods spread in industry.

Keywords: integrated design, design optimization, design automation, computational thinking, professional skills, engineering education

Résumé

L'intelligence artificielle a un impact considérable sur de nombreux domaines dans nos sociétés. L'ingénierie mécanique n'est pas épargnée et voit l'apparition de nouvelles techniques et de nouveaux procédés. La formation des futurs ingénieurs doit s'adapter à cette nouvelle réalité, en changeant les techniques, mais aussi les compétences enseignées.

En utilisant la conception d'actionneurs électromécaniques comme fil conducteur, ce travail explore les nombreuses facettes de la conception automatisée de systèmes : modélisation, optimisation et éducation. Il recherche en particulier les conditions nécessaires à une utilisation réussie.

Le travail commence par la construction d'un modèle intégré modulaire. Les prédictions des spécifications au niveau système sont privilégiées puisqu'elles apportent une forte valeur ajoutée pour les décideurs et réduisent la distance entre le modèle et le produit fini. Un outil de conception automatisé est créé en combinant ce modèle à des algorithmes évolutifs multiobjectifs et à des outils de visualisation des résultats. Cet outil vise à présenter rapidement aux ingénieurs et aux décideurs des informations pertinentes sur leur travail de conception. Le potentiel et les avantages de l'outil sont validés par deux exemples concrets. Toutefois, les résultats mettent également en évidence un écart dans les performances des algorithmes d'optimisation entre des problèmes de référence et des problèmes de conception.

Afin d'améliorer ces algorithmes, des problèmes de référence appropriés et réalistes sont nécessaires. À cette fin, un sous-ensemble du modèle de conception intégrée est utilisé pour formuler une nouvelle suite de tests appelée MODAct. Cette dernière est composée de 20 problèmes d'optimisation multiobjectifs sous contraintes avec des niveaux de complexité variables. En outre, des méthodes numériques pour évaluer la morphologie de l'espace de recherche sont formulées et utilisées pour identifier les différences entre les caractéristiques de MODAct par rapport à 45 problèmes de référence tirés de la littérature. De plus, les performances de convergence de trois algorithmes sur ces mêmes problèmes mettent en évidence le rôle clé des contraintes, et en particulier, le nombre de contraintes simultanément violées dans MODAct.

Dans un second temps, une nouvelle technique de gestion des contraintes adaptée aux algorithmes évolutifs et pensée pour les problèmes fortement contraints est décrite. Elle est comparée à de nombreuses stratégies existantes de gestion des contraintes. Pour s'assurer des meilleures performances, les paramètres de toutes ces méthodes sont ajustés selon une procédure de méta-optimisation. La comparaison de la convergence des diverses configurations met à nouveau en évidence une différence entre MODAct et les problèmes de référence. De plus, la nouvelle méthode représente le meilleur compromis en matière de performance sur l'ensemble des problèmes d'optimisation considérés.

À mesure que les limitations techniques sont supprimées, le travail des ingénieurs ainsi que la formation des futurs ingénieurs se retrouvent impactés par la conception automatisée. D'une part, le développement de compétences dites professionnelles des étudiants à travers leur travail sur deux types de projets d'équipe a été évalué grâce à 205 étudiants de trois cours. Leur apprentissage était plus solide et plus complet quand ces compétences étaient abordées explicitement en cours. Ces résultats suggèrent qu'il est possible de favoriser l'apprentissage de compétences transversales sans forcément revoir complètement les programmes d'études. D'autre part, neuf groupes (33 étudiants) ont été invités à concevoir un actionneur en utilisant d'abord une approche conventionnelle, puis des méthodes de conception automatisée. Les actionneurs suggérés par les étudiants issus de l'outil automatisé surpassent les solutions obtenues par l'approche traditionnelle. Six groupes proposent même des solutions moins coûteuses, dont trois sont également plus petites, que le produit industriel, conçu par des ingénieurs expérimentés. Les étudiants ont prouvé qu'ils étaient capables d'utiliser le potentiel de l'outil en un temps relativement court. L'analyse de leurs erreurs suggère des améliorations possibles pour de futurs outils. En quittant l'université, ces étudiants emportent avec eux l'espoir de voir de telles méthodes s'installer dans l'industrie.

Mots-clés : conception intégrée, conception optimale, conception automatisée, pensée computationnel, compétences transversales, formation des ingénieurs

List of publications and codes

Publications

- V. Mounier, C. Picard, and J. Schiffmann. "Data-Driven Pre-Design Tool for Small Scale Centrifugal Compressors in Refrigeration". In: *Journal of Engineering for Gas Turbines and Power* (July 27, 2018). DOI: 10.1115/1.4040845
- C. Picard and J. Schiffmann. "Impacts of Constraints and Constraint Handling Strategies for Multi-Objective Mechanical Design Problems". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '18. New York, NY, USA: ACM, 2018, pp. 1341–1347. DOI: 10.1145/3205455.3205526
- C. Picard and J. Schiffmann. "Realistic Constrained Multiobjective Optimization Benchmark Problems From Design". In: *IEEE Transactions on Evolutionary Computation* 25.2 (Apr. 2021), pp. 234–246. DOI: 10.1109/TEVC.2020.3020046
- C. Picard and J. Schiffmann. "Automated Design Tool for Automotive Control Actuators". In: *IDETC-CIE2020*. ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 11B: 46th Design Automation Conference (DAC), Aug. 17, 2020. DOI: 10.1115/DETC2020-22390
- C. Picard, C. Hardebolle, R. Tormey and J. Schiffmann. "Which Professional Skills do Students Learn in Engineering Team-Based Projects?". *Accepted for publication in the European Journal of Engineering Education*, April 2021.

Data and code

- C. Picard and J. Schiffmann. *Multi-Objective Design of Actuators: Pareto Fronts*. Zenodo, May 13, 2020. DOI: 10.5281/zenodo.3824302
- MODAct: https://github.com/epfl-lamd/modact
- Python wrapper for the hypervolume by WFG: https://github.com/epfl-lamd/hvwfg
- Contributions to https://github.com/msu-coinlab/pymoo

Contents

Acknowledgements i

Abstract (English/Français) iii

List of publications and codes vii

List of Figures xiii

List of Tables xix

Nomenclature xxiii

1 Introduction 1
- 1.1 Background . . . 2
  - 1.1.1 Design and optimal design . . . 2
  - 1.1.2 Multiobjective optimization . . . 4
  - 1.1.3 Automated design . . . 6
  - 1.1.4 Engineering education . . . 7
- 1.2 Electro-mechanical actuators . . . 8
- 1.3 Problem statement . . . 10
- 1.4 Goals and objectives . . . 10
- 1.5 Outline of the thesis . . . 11

2 Automated Design of Electro-mechanical Actuators 13
- 2.1 Building an integrated model . . . 14
- 2.2 Component models . . . 15
  - 2.2.1 Stepper motor . . . 16
  - 2.2.2 Spur gears . . . 20
- 2.3 System-level models . . . 21
  - 2.3.1 Operating conditions . . . 22
  - 2.3.2 Packaging . . . 23
  - 2.3.3 Assembly feasibility assessment . . . 30
  - 2.3.4 Shafts supported by the housing . . . 33
- 2.4 Automated Design Tool . . . 35

2.4.1 Overview 35
2.4.2 Defining the problem 35
2.4.3 Multiobjective optimizer 38
2.4.4 Interactive result visualization 40
2.5 Case studies 40
2.5.1 HVAC Valve Actuators 42
2.5.2 HVAC flaps actuators 45
2.6 Optimization challenges 47
2.7 Concluding remarks 49

3 Realistic Benchmark Problems 51
3.1 Related work 52
3.1.1 Constrained multiobjective optimization benchmark problems . . . 52
3.1.2 Constraint landscape analysis 53
3.2 Multi-Objective Design of Actuators (MODAct) 56
3.2.1 Definition of the various problems 56
3.3 Multiobjective constraint landscape analysis 59
3.4 Methods of the numerical investigations 61
3.4.1 Constraint landscape analysis 61
3.4.2 Convergence study 61
3.5 Results and discussion 63
3.5.1 Design trade-offs and constraints 63
3.5.2 Convergence analysis 64
3.5.3 Link between convergence and constraints 67
3.6 Concluding remarks 73
3.6.1 Key messages 73
3.6.2 Creating variants of the proposed MODAct instances 74

4 An Improved Constraint Handling Strategy: cEpsilon 77
4.1 Background 78
4.1.1 Overview 78
4.1.2 Selected ϵ constraint-handling methods 80
4.2 Adapted per-constraint epsilon values 83
4.3 Methods of the numerical investigations 83
4.3.1 Integration within NSGA-III 84
4.3.2 Experimental conditions and data analysis 85
4.3.3 Parameter tuning setup 87
4.4 Results 87
4.4.1 Automated tuning of parameters 87
4.4.2 Performance comparison 87
4.5 Concluding remarks 96
4.5.1 Discussion 96
4.5.2 Path forward 96

5 Educating Future Engineers 99
5.1 Context of the studies 100
5.1.1 Involved courses and study design 100
5.1.2 Data collection protocol and population 101
5.2 Professional skills' development through projects 102
5.2.1 Related work 103
5.2.2 Methodology 108
5.2.3 Results 110
5.2.4 Discussion 114
5.2.5 Concluding remarks 117
5.3 Use of machine intelligence by novice engineers 118
5.3.1 Study design 118
5.3.2 OMOD – User interface 120
5.3.3 Approach to the analysis of data 121
5.3.4 Impact on the selected products 123
5.3.5 Understanding the process 127
5.3.6 Discussion 136
5.4 Concluding Remarks 138

6 Conclusion 141
6.1 Summary 141
6.2 Outlook 143

A Comparison of the steady-state stepper motor model 145

B Interprofessional project management questionnaire (IPMQ) 147

C OMOD: Example Files 151

Bibliography 157

Curriculum Vitae 177

List of Figures

1.1 Comparison of (A) the ‘Systematic Approach’ by Pahl et al. [134] with (B) a schematic representation of its application in practice. 2

1.2 Introductory example to multiobjective optimization showing the trade-off between price and portability for some generic smartphone, tablet, laptop, and desktop computer. 4

1.3 Proposed updated design process to include automated design steps. 7

1.4 Examples of automotive electro-mechanical actuators—courtesy of Johnson Electric (JE). 9

2.1 Multiple ways to make one actuator, actuator image courtesy of JE. 13

2.2 Schematic representation of the way individual components are connected to form an integrated model. 15

2.3 Comparison between the speed-torque characteristics obtained through the simulation of the ODEs and through the steady-state model, highlighting the effect of different features: (A) stepper with friction, (B) adding saturation and (C) adding a current limitation. 19

2.4 Representation of the available mesh geometries for the motors and their parameters. 19

2.5 Planar view of the special geometry used to model a gear pair: with respect to the real gear geometry (left) or with the key variables (right). 21

2.6 Schematic representation of an actuator with a crossed helical gear stage and two spur gear stages, showing the relative positioning procedure of the components, and the resulting bounding box (side and top views). 24

2.7 Location of the positive and negative hooks for spur gears and crossed helical gears in forward (left) and backward (right) mode 25

2.8 Comparison of the resulting actuator layout for the forward (left) and backward (right) modes, for an actuator composed of a stepper motor and two spur gear pairs. 26

2.9 Illustration of an actuator with and without internal collision. Colliding faces detected by FCL are shown in red. 26

2.10 Examples of possible hulls for actuators where the black dot illustrates the position of the origin of the reference frame and corresponds to the required position of the output shaft. The volumes are obtained by extruding these shapes. 27

2.11 Evolution of the constraint value for the inclusion in the hull as the actuator is rotated around the first gear pair by an angle $\theta = \pi/2$. 29

2.12 Comparison of the naive versus selective method in terms of execution speed for the example shown in Figure 2.11—calculations done on a 2.7 GHz *Quad-Core Intel Core i7* laptop, number of vertices: 96 for the hull and 252 for the actuator. 30

2.13 Two four-stage actuators examples—(A) a step-like actuator layout and (B) the interlaced actuator layout—and their respective assembly dependency graph, where the shaft groups are numbered successively. Each arrow indicates a dependency. 31

2.14 (A) Example of an actuator with a crossed helical gear stage and two spur gear stages showing the rays of the assembly evaluation procedure. The group in red corresponds to ②, which can only be inserted after the two gear shafts. (B) The resulting assembly dependency graph, where each arrow indicates a dependency. The motor is not connected, indicating it has no requirement and is not a dependency for any other component. 32

2.15 Example of a four-stage actuator with an assembly deadlock. 32

2.16 Illustrations of the method used to detect if shafts can be supported by the housing for two actuators. Collisions between the rays and gears not belonging to the corresponding shaft group are shown in red. 33

2.17 Evolution of the pass-through constraint value with and without the “shaft overlap” feature for a two-gear-pair system—shown in top and side views—where the top gear pair is rotated from an initial position, where all axes are aligned, to an angle $\theta = \pi/3$. 35

2.18 Overview of the working principle of the automated design tool for actuators. 36

2.19 Screenshot of the HTML reporting interface. 41

2.20 Acceptable hull for the HVAC valve actuator application with the position of the output shaft shown as a black dot along with the simplified contour used for the optimizations as a dashed line. 42

2.21 Comparison of the proposed designs in terms of total cost and torque excess between an optimization without and with system-level constraints: (A) entire range and (B) zoomed view. The calculated properties of an existing actuator are shown as well. 43

2.22 Example of a design returned by the tool when not considering layout constraints. 44

2.23 Comparison of two alternate designs that match the requirements for an HVAC valve actuator using (A) a DC motor or (B) a stepper motor. 44

2.24 Comparison between the obtained Pareto fronts for HVAC flap actuators with three, four, and five stages of spur gears. 46
2.25 Comparison of the layout of (A) the existing flap actuator and (B) a candidate actuator generated by the automated design tool. 46
2.26 (A) Pareto front of the flap actuator design problem when the shaft pass-through constraint is set as an objective and (B) an example of a compact actuator design. 47
2.27 Picture of a 3-D printed candidate design. 48
2.28 Fronts obtained for the HVAC flap actuator design problem for four different optimization runs. 48

3.1 Best-known Pareto fronts for (A) CS and (B) CT problems 63
3.2 Best-known Pareto fronts for CTS problems 64
3.3 Best-known Pareto front for CTSEI3, for rendering questions, the numbers of points displayed has been reduced by eliminating points that were too close to each other 65
3.4 Box plot of the relative hypervolume errors ΔHV_n of the external archives obtained for each problem by NSGA-II/NSGA-III versus C-TAEA, where the $*$ or † after the function name is used to indicate that C-TAEA is, respectively, significantly worse or better than NSGA-II/NSGA-III 66
3.5 Comparison of the approximate Pareto fronts obtained by the best runs for NSGA-II and C-TAEA and two runs achieving a ΔHV_n of 0.9 and 0.5 against the best-known Pareto front for CS4 67
3.6 Evolution of the median relative hypervolume error including 5th and 95th percentiles for CT problems comparing: NSGA-II with the constrained-dominance strategy (CDP-NSGA-II), C-TAEA and NSGA-II discarding constraints (Unconstrained NSGA-II) 68
3.7 Map of the problems based on their (A) feasability ratio (FsR)-normalized ratio of feasible boundary crossing (nRFB$_\times$) and (B) PFd-PFcv scores grouped by constraint levels or test suites 70
3.8 Color-maps of the normalized constraint violation values as a function of the objective values based on the random samples in $\mathcal{L}$ for CT3, CF7, DAS-CMOP3_12 and MW11 with their respective (best-known or true) Pareto front in red dots. For the generation of the color-map, the constraint violation values of points very close to each others in the objective space are aggregated using the lowest value 72
3.9 (A) Relative share of samples from $\mathcal{L}$ and (B) average share of evaluated solutions by NSGA-II or NSGA-III by number of simultaneously violated constraints for all considered problems with more than two constraints 73

4.1 Evolution of ϵ along generations for different parameter values. 81
4.2 Comparison between the best and median final populations obtained by aE, cE_1, cE_2, and iE_1 and the best-known Pareto fronts for CS4 and CT4. 94

4.3 Evolution of the mean ϵ value in logarithmic scale across all optimization runs (and if applicable, across the multiple ϵ_j) for four configurations on CS4 and CT4. 95

4.4 Evolution of the mean individual ϵ_j values in logarithmic scale across all optimization runs with cE_2 and ipE_1 of CS4 and CT4. 95

5.1 Overview of the timeline for the three courses highlighting the different phases and when the questionnaires were collected. 101

5.2 Cross-sectional comparison of pre-scores of students and professionals per factor and of the total score of the IPMQ. The 1.5 interquartile range (IQR) convention is used for the whiskers and the outlier identification. In addition, the mean value of the distributions is shown with a red triangle. Statistically significant differences between populations within factors are shown using the star notation, see Table 5.3. 110

5.3 Box plot comparing pre/post scores of BA students per factor (mean value indicated by a red triangle). Only students that answered both questionnaires are considered ($N = 47$). Statistically significant differences within factors are shown using the star notation, see also Table 5.4. 112

5.4 Box plot comparing pre/post scores of MA students per factor (mean as a red triangle). Only students that answered both questionnaires are considered ($N = 29$). Statistically significant differences within factors are shown using the star notation, see also Table 5.8. 116

5.5 Overview of the models accessible to the students in each phase. 118

5.6 General architecture overview of the OMOD interface. 121

5.7 Comparison of the proposed actuators based on selected motor, number of stages and the use of multiple materials for the gears. 123

5.8 Paired plots comparing actuators between the two phases based on (A) specifications and (B) performance objectives. All individual solutions are represented as dots and colored in green and red to indicate an increase or a decrease from the traditional to the automated design respectively. Grayed zones indicate specification violation zones. The industrial product is shown as an olive point. 124

5.9 Biobjective comparison of the actuators where colors represent each group (GID) and marker styles each phase. 125

5.10 Comparison of three gear parameters by stages between traditional and automated design. Individual points are represented by dots. Since not all designs have the same number of stages, the number of points varies by stage. The parameters of the gears of the industrial product are shown in olive. . 126

5.11 Heat map of the number of simulations per day and per group. The shown time window starts from the start of OMOD to report 2. The colorbars are cropped at the 98th (A) and 99th (B) percentiles. The ticks of the x-axis correspond to the days on which the weekly lectures or meetings were scheduled. The blue and red bars indicate for each group the start and the end of the search process, meaning that there are more than 4 (A) or 5 (B) consecutive simulations with less than an hour in between each. (A) displays the motor simulations, and for each group distinguishes between the simulations of the same motor kind “S” and the others “O”. The orange dots show the first time the exact motor configuration selected in report 2 is evaluated. (B) displays the gear simulations. The colored dots show the first time a stage of the actuator of report 2 is simulated. 130
5.12 Heat map of the number of submitted optimizations per day, per group and per number of gear stages (showing only $\geq$ 3). The shown time window spans the automated design phase. The oral presentation is shown as a brown bar. The colorbar is cropped at the 98th percentiles. The ticks of the x-axis correspond to the days on which the weekly lectures or meetings were scheduled. The blue and red bars mark the first and last day when optimizations that are not the provided example are executed. The orange dots show when the optimization from which the final design stems is performed. The pink dots and diamonds show when the actuator in or the closest to report 3 has been simulated in the actuator module (first appearance). The green and purple dots indicate when the actuator of report 2 and the first actuator linked to an optimization has been simulated in the actuator module. 132
5.13 Per day view between the start of the automated design phase and the final presentation. (A) Number of optimizations separated based on their inclusion in a batch. The daily means for each week are shown as a solid line. (B) Number of optimizations colored by the number of attributes (e.g., chosen objective combinations or constraints) they share with the given example file. (C) Number of actuator simulations originating from an optimization colored depending on how they respect the specifications. 135
5.14 Map of all valid actuators originating from an optimization based on their relative cost and relative volume, and colored by group (GID). 135
5.15 Anonymous student feedback for the MA course ($N = 30$). 137

A.1 Comparison between stepper models for all defined steppers at $V_s = 12\,\mathrm{V}$ and with no current limitation. 146

List of Tables

2.1 List of crossover and mutation operators used in the adaptive variation operator. 38
2.2 Set of operating points for the HVAC valve application. 42
2.3 Set of operating points to evaluate HVAC flap actuators. 45

3.1 Number of objectives, search variables and constraints of the selected benchmark constrained multiobjective optimization problems (CMOPs) along with their FsR—calculated as explained in Section 3.3 ©IEEE 2021 54
3.2 Operating point requirements for all Multi-Objective Design of Actuators (MODAct) problems ©IEEE 2021 57
3.3 Summary of the 20 functions of MODAct including details about their constraints and the search space ©IEEE 2021 58
3.4 Parameters used to configure each run of NSGA-II, NSGA-III and C-TAEA ©IEEE 2021 62
3.5 Calculated metrics resulting from the constraint landscape analysis ©IEEE 2021 69

4.1 Number of objectives, decision variables and constraints of the selected benchmark CMOPs and the defined number of generations for each problem. . . 86
4.2 Parameters used to configure each run of NSGA-III. 86
4.3 Parameters and their range for the parameter tuning configuration of irace. 87
4.4 List of the considered constraint handling strategies (CHSs) with the parameters of the various configurations obtained through the tuning procedure. 88
4.5 Median and interquartile range (IQR) of the hypervolume for the obtained Pareto fronts for each problem and CHS configuration. The cell with the highest median value per problem is highlighted in gray. 89
4.5 (Continued) 90
4.5 (Continued) 91
4.6 Results from the statistical comparison procedure summarizing for each method and problem how many times the method performed better/equal/worse than the others. For problems where the null hypothesis of the Kruskal-Wallis test could not be rejected, all methods are considered equivalent. Best performing methods are highlighted in gray. 93

4.7 Mean rank for each CHS per problem family. Rank is dense and ties are given the same rank (smaller is better). Best ranked CHSs are marked in bold. . . 94

5.1 Summary of the different participants and the data collected. 108
5.2 Mean and standard deviation (SD) of students' pre-scores and professionals' scores per factor of the IPMQ. 110
5.3 Results of the independent T-test statistical pairwise comparison between populations (BA, MA and Pro) within factors of the IPMQ with the alternative hypothesis that scores improve from BA to Pro (one-sided tail). 111
5.4 Mean and standard deviation, and results of the paired T-tests between pre/-post scores per factor for BA students with the alternative hypothesis that post scores are better (one-sided tail). Only students that answered both questionnaires are considered ($N = 47$). 112
5.5 Reported prior experience and training by BA students and evolution during the semester. 113
5.6 Analysis of students' answers to the open question "The 3 most important non-technical things I learned while carrying out the team project are:", N=136. The table presents only answers with more than 4 occurrences, grouped by topic with verbatim examples and ordered by decreasing frequency. 114
5.7 Analysis of students' answers to the open question "The 2 biggest challenges I encountered during the realization of the team project:", combined with two Likert questions: "For the [first/second] of these challenges, I learned the skills to better manage a similar problem in the future". The number of students having answered all three questions is N=110. Answers are grouped by topic, presented with examples and ordered by decreasing number of occurrences on "Agree". Answers with 4 occurrences or fewer are not presented. 115
5.8 Mean and standard deviation (SD), and results of the paired T-test statistical comparison of pre/post IPMQ scores per factor for MA students with the alternative hypothesis that post scores are better (one-sided tail). Only students that answered both questionnaires are considered ($N = 29$). 117
5.9 Relevant specifications for the actuator to be designed. 119
5.10 Most critical operating points based on Table 5.9 used to evaluate the actuators and assess their performance. 122
5.11 GID of the groups whose design violates certain specifications. 125
5.12 Number of unique simulations by available model per student. 128
5.13 Number of optimizations by category (see Section 5.3.3). 131
5.14 Number of different settings evaluated by students in their optimizations, excluding example optimizations. 133
5.15 Number of times similar optimizations—differing solely in terms of optimization settings (budget or search bounds)—have been performed. 133
5.16 Overview of the optimizations from which the actuators in report 3 have been chosen and the differences. 136

B.1 List of the questions of the IPMQ in English and *in French* along with the factors they are linked to (A=planning, B=risk assessment, C=ethical sensitivity, D=communication, E=interprofessional competence). 147
B.1 (continued) 148
B.1 (continued) 149

Nomenclature

Symbols

α	Blend crossover coefficient
α	Pressure angle
α_{el}	Temperature coefficient for the resistance
α_{mag}	Temperature coefficient for the magnetic flux
ΔHV_n	Relative hypervolume error
ΔT	Torque excess
ϵ	Relaxation value
ϵ_γ	Total contact ratio
η	Conversion efficiency
η_c	Crossover coefficient
η_m	Mutation coefficient
γ	Angle between components
γ	Fraction of the population
λ	Fraction of the maximum number of generations
$\mathcal{A}$	Set of solutions in an external archive
$\mathcal{F}$	Feasible part of the search space, $\mathcal{F} \subseteq \mathcal{S}$
$\mathcal{L}$	Set of all randomly sampled points
$\mathcal{PF}^*$	Pareto front
$\mathcal{PF}_n^*$	Normalized Pareto front
$\mathcal{P}^*$	Pareto optimal set
$\mathcal{S}$	Search space
$\mathcal{U}$	Uniform sampling
$\mathcal{W}$	Random walk
μ	Population size in evolutionary algorithms
ω	Rotational speed
$\bar{c}$	Cost per weight
ϕ_0	Linkage flux
ρ	Density

ρ^*_{fp}	Root radius factor
σ	Stress
θ	Angle
θ	Index of an individual solution
$\boldsymbol{f}$	Objective function vector
$\boldsymbol{f}^n$	Normalized objective function vector
$\boldsymbol{g}$	Inequality constraint vector
$\boldsymbol{h}$	Equality constraint vector
$\boldsymbol{r}$	Reference vector for the hypervolume calculations
$\boldsymbol{x}$	Decision or design vector
$\boldsymbol{x}^*$	Optimal decision or design vector
$\boldsymbol{x}^{(L)}$	Lower limit for the decision vector
$\boldsymbol{x}^{(U)}$	Upper limit for the decision vector
$\boldsymbol{z}$	Objective value vector
$\boldsymbol{z}^*$	Ideal vector
$\boldsymbol{z}^{\mathrm{nad}}$	Nadir vector
ζ	Specific sliding
A_h	Surface of the actuator housing
b	Gear thickness
b^*	Relative gear thickness, $b^* = b/m$
bb_y	Dimension of the bounding box in direction y
bb_z	Dimension of the bounding box in direction z
c	Cost
c_p	ϵ relaxation control coefficient
CV	Constraint violation
d	Cohen's d
d	Diameter
d	Distance between components
d_w	Wire diameter
f_i	i^{th} objective function
f_{dyn}	Viscous friction coefficient
f_{el}	Temperature correction factor for the resistance
f_{mag}	Temperature correction factor for the magnetic flux
FF	Fill factor of the windings
g_j	j^{th} inequality constraint
h	Thickness
h^*_{ap}	Addendum coefficient

h_{fp}^*	Dedendum coefficient
h_k	k^{th} equality constraint
HV	Hypervolume
I	Current
i	Transmission ratio
$I_{\max}$	Maximum admissible current
K	Number of neighbors
k_m	Motor constant
k_s	Current correction factor
k_{m0}	Nominal motor constant
L	Inductance
l	Length
m	Gear module
mFF	Combined motor index and fill factor
N_w	Number of turns in the windings
p	Number of inequality constraints or number of rotor teeth
p_c	Crossover probability
p_f	Stochastic ranking probability
p_m	Mutation probability
R	Resistance
r	Radius
R_{scale}	Resistance scaling factor
S	Safety factor
T	Torque
T_c	Coulomb friction torque
T_c	Cut-off generation
T_m	Motor torque
T_s	Temperature of the system
$T_{\max}$	Maximum number of generations
T_{po}	Pull-out torque
V	Voltage
V	Volume
x	Cartesian coordinate
x	Profile shift coefficient
y	Cartesian coordinate
Z	Number of teeth
z	Cartesian coordinate

Zx	Combined Z and x variables

Operators

$\lvert\cdot\rvert$	Absolute value or magnitude
$\cdot \preceq \cdot$	Pareto dominance preference relation
$\cdot \preceq_{\epsilon} \cdot$	Preference relation with ϵ relaxation
$\langle\cdot,\cdot\rangle$	Inner product
$\lvert\lvert\cdot\rvert\rvert$	Euclidean norm

Subscripts

1	Index of the pinion in a gear pair
2	Index of the wheel in a gear pair
max	Maximum
min	Minimum
harm	Harmonic
a	Addendum (tooth tip)
eq	Equivalent
F	Tooth root (stresses)
f	Dedendum (tooth root)
H	Tooth flank
m	Motor
nom	Nominal conditions at simulation

Allons, à mes dépens
je vois que l'on veut rire.
Il en peut coûter cher. Eh bien! soit.
J'y consens.

Raoul (Les Huguenots, G. Meyerbeer)

1

Introduction

MACHINE INTELLIGENCE[1]—also referred to as *artificial intelligence* (AI)—has seen a sharp extension of its capabilities and its outreach to the general public over the past decade. Not a single day passes without announcements of the launch of new AI-powered tools: from trivial photo filters in social media apps to conservation and monitoring of forests—e.g., the TreeSatAI project by TU Berlin [180]—to diagnostic and prognosis tools for healthcare practitioners —e.g., [74, 88, 177, 196]. Initially thought to be limited to the automation of routine tasks, machine intelligence is extending into creative domains [21, 22, 69, 163, 164]. Its impact could be deep and affect the kind of skills required in numerous professions including the engineers [57].

The AI revolution has also hit the traditionally conservative field of mechanical engineering [129]. In combination with AI, advanced manufacturing—among which 3-D printing—robotics, and electronics are transforming the way people interact, build or create products. For example, electric and autonomous cars have disrupted the automotive industry, once driven by purely mechanical considerations [4]. The electricity market once dominated by rotating conversion machines sees an uptake of different technologies using photovoltaic and other electrochemical processes.

The promise of these revolutions is to be able to solve increasingly complex problems and tackle the challenges posed for example by climate change. Pushing systems to their limits and improving their efficiency requires considering the systems as a whole and breaking the barrier of disciplines [159]. Especially since humans struggle to cope with coupled effects and lack system-level awareness [199], machine intelligence seems capable to open up new perspectives. However, for these opportunities to be successful, all their ingredients need to match and engineers should take the risk to seize the moment.

---

[1] Machine intelligence is used here as a generic concept describing all kinds of uses of computer programs to perform *intelligent* tasks. The definition of *intelligence* is purposely omitted in this work, since it is debated for humans [55, 168], let along machines [120].

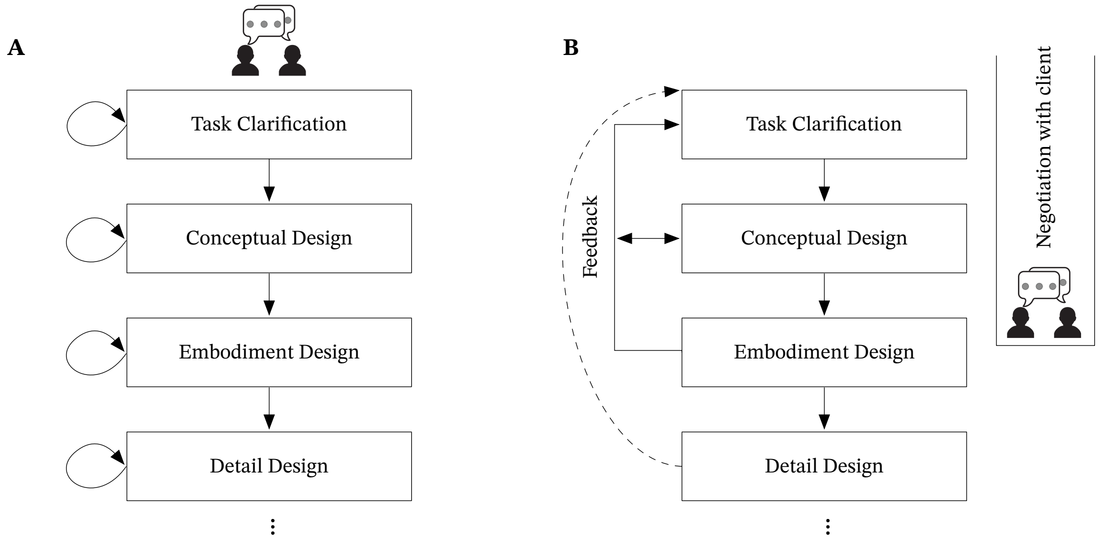


Figure 1.1: Comparison of (A) the 'Systematic Approach' by Pahl et al. [134] with (B) a schematic representation of its application in practice.

1.1 Background

1.1.1 Design and optimal design

Design is a complex decision-making process whose aim is the selection of a set of suitable entities to form a system capable of performing a function [137]. While some see it as a mysterious and creative process only, design has and is being extensively studied [10], and several authors suggested descriptions of the process and recommendations [44, 62, 134, 181]. In the German engineering school [134] for example, this process undergoes several steps: 1. task clarification, 2. conceptual, 3. embodiment, and 4. detail design, see Figure 1.1(A). The first step aims at understanding and refining the design task and the associated specifications. The conceptual design step is a divergent thinking step where creative working principles that could fulfill the task are sought. The embodiment and detailed design phases are successive convergent steps, where concepts are put into physical form, refined and detailed up to having all production documents. These steps should, in theory, be performed in order, with iterations only within each step [92].

Such description seems to suggest an overly simplistic linear process, which is far from realistic. Decisions taken throughout the stages have sometimes complex implications, requiring repeated feedback loops to progress. The role of the stakeholder or client is also neglected, implying that the requirements are to be settled before the design process starts. Yet, in practice, discussions with the client move along the process and can in turn trigger loops as specifications are clarified and refined, see Figure 1.1(B). Others go further and lay the analysis at the center of the process since it informs every step [75].

Since design is a decision-making process, it is legitimate to look into how, ideally informed, decisions are taken. Traditionally, after generating—synthesizing—several physical configurations, their performances with respect to the specifications are evaluated. This evaluation can be done through prototypes [90] or based on engineering science and its models [137]. Once knowledge is gained about the configurations, the "best" option is selected. While the concept might seem intuitive, there a few subtleties behind what is "best", and its formal definition forms the field of *optimal design*.

Papalambros and Wilde [137] define design optimization "as the selection of the 'best' design within the available means". When designs can be described by a vector of n design variables $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$, design optimization can be formally transposed into a mathematical optimization problem, arbitrarily defining the "best" as the one having the smallest objective value:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{S}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_j(\boldsymbol{x}) \leq 0, \quad j = 1, 2, \dots, p \\ & h_k(\boldsymbol{x}) = 0, \quad k = 1, 2, \dots, q \end{aligned} \tag{1.1}$$

where f is the selection criterion or objective, $\mathcal{S}$ the design space, and the "available means" are described by p inequality and q equality constraints. Feasible or valid designs satisfy all $p + q$ constraints, and the set of all feasible solutions forms the feasible space $\mathcal{F} \subseteq \mathcal{S}$.

Given that one is capable of formulating numerical objectives and constraints, finding the optimal design comes down to solving the optimization problem Eq. (1.1). While designs could also be evaluated experimentally with prototypes, this definition suggests nowadays the use of computer models. The latter are very convenient since they allow the prediction of many designs without having to build the real systems. Models are an abstraction and are generally derived based on assumptions and care must be given to find a good balance between modelling level, model computation speed, and accuracy.

One might be tempted to go for the most detailed models possible. The caveat is, however, that one should consider the final goals of design optimization. In this context, the numerical constraints and objective serve to order the design and select the "best". So, it is the correct delimitation of the feasible space and the order of the solutions that is paramount. Detailed models are often complex, computationally expensive, and therefore, currently, limited in their scope. This often leads to a segregated design approach: components of a system are considered and optimized individually. Multiple iterations are then required to converge on the whole product [197]. Further, this approach does not guarantee that an optimal system will be obtained when sub-components are coupled—i.e., they depend on each other, which is often the case for mechanical systems. Indeed, Schiffmann [159] showed that the seasonal efficiency of a heat pump system could be increased by 12 points when, instead of a segregated approach, an integrated design approach was followed. Integrated models

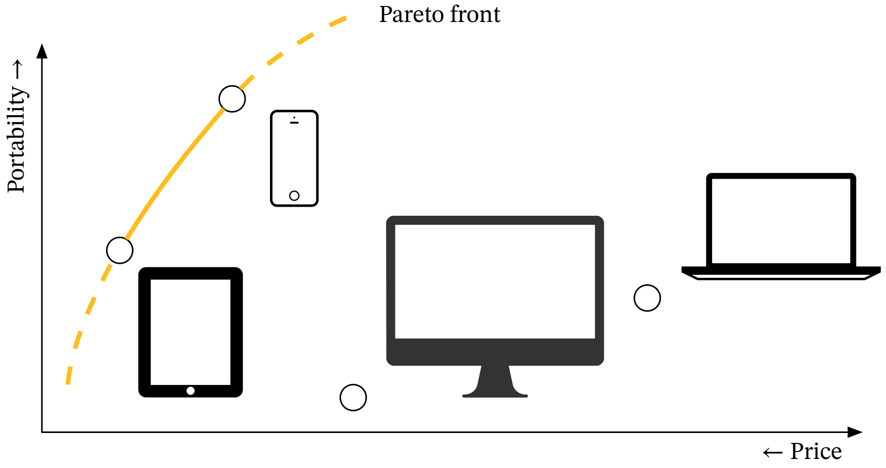


Figure 1.2: Introductory example to multiobjective optimization showing the trade-off between price and portability for some generic smartphone, tablet, laptop, and desktop computer.

consider a system as a whole, and since design task specifications often express constraints at the system-level, fewer assumptions are needed to ensure that a valid system is found.

1.1.2 Multiobjective optimization

As an example, consider the selection of a smart device using the principle of optimal design. The available options are a smartphone, a tablet, a laptop and a desktop computer. The price of the device is often an important criterion. Assuming the prices of these devices are CHF 1200, CHF 700, CHF 2200 and CHF 1500, respectively, it would mean that, in this example, the tablet should be selected. It is, however, obvious that such devices cannot be compared solely by price. Considering another criterion such as portability, the best device would probably be the smartphone. If one wanted to optimize both objectives, on could for example combine the objectives using weights:

$$f(\boldsymbol{x}) = w_1 \cdot \mathrm{cost}(\boldsymbol{x}) + w_2 \cdot \mathrm{portability}(\boldsymbol{x}) \tag{1.2}$$

This approach is simple, but selecting the weights is difficult. If both objectives are equally important, one could set $w_1 = w_2 = 0.5$, but the price and portability have different scales and units. If portability is a score between 0 and 1, and both objectives are conflicting with each other, these weights would actually favor the price.

Instead of deciding the trade-off through weights *a priori*, another approach is to consider both objectives simultaneously through a multiobjective optimization (MOO) approach. Figure 1.2 shows how in the example, the devices are spread in a price-portability map, the objective space in this case. From it, it seems clear that in such a case the smartphone and the tablet are both optimal solutions representing different trade-offs, while the laptop and the desktop computer are so-called dominated solutions.

More formally, in a MOO setting, the optimization problem Eq. (1.1) is rewritten as:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{S}} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_m(\boldsymbol{x})]^T \\ \text{subject to} \quad & g_j(\boldsymbol{x}) \leq 0, \quad j = 1, 2, \dots, p \\ & h_k(\boldsymbol{x}) = 0, \quad k = 1, 2, \dots, q \end{aligned} \tag{1.3}$$

where now $\boldsymbol{f} : \mathcal{S} \rightarrow \mathbb{R}^m$ is a vector of $m > 1$ objective functions. MOO compares solutions using the concept of *Pareto optimum* [45, 138], which leads to the notion of Pareto dominance [26].

Definition 1.1 (Pareto dominance)**.** $\boldsymbol{x}_1$ is said to weakly Pareto dominate $\boldsymbol{x}_2$, written $\boldsymbol{x}_1 \preceq \boldsymbol{x}_2$, if and only if $\forall i \in \{1, \dots, m\}, f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$. $\boldsymbol{x}_1$ is said to Pareto dominate $\boldsymbol{x}_2$, written $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$, if in addition, there exists one $i \in \{1, \dots, m\}$ for which $f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2)$.

The optimum of a multiobjective optimization problem becomes thus a set of solutions: the *Pareto optimal set* $\mathcal{P}^*$, which is the set of all non-dominated solutions, i.e.,

$$\mathcal{P}^* := \{\boldsymbol{x}^* \in \mathcal{S} \mid \neg\exists \boldsymbol{x} \in \mathcal{S} : \boldsymbol{x} \prec \boldsymbol{x}^*\} \tag{1.4}$$

The image of $\mathcal{P}^*$ in the objective space is called the Pareto front $\mathcal{PF}^*$.

Engineers often have to make compromises between conflicting interests in order to find a solution most suitable with respect to the specifications. MOO is ideal to handle such situations, since it gives insights into the trade-offs within a problem. As such, it promotes informed decision-making.

In general, however, solving the global optimization problem (1.3) is difficult. Engineering problems can be particularly challenging, since the objectives and constraints can be computationally expensive to calculate, offer no guarantee about continuity or convexity, and their behavior cannot be inferred *a priori* from equations—at least for simulation-based models. For such problems, diverse stochastic meta-heuristics have been successfully applied [26]. Contrary to most deterministic optimizer derived from gradient-descent methods, stochastic meta-heuristics are biased random search algorithms and have therefore minimal requirements about the nature of the optimization problem. Their downside is their convergence, which is not always repeatable. While some algorithms have a theoretical convergence guarantee at $t \rightarrow \infty$, this is not the case for all algorithms and depends on the chosen heuristics [7].

Among all the meta-heuristics for MOO, the field of evolutionary computation has been extremely active in recent years proposing numerous multiobjective evolutionary algorithms (MOEAs). Some examples include evolutionary strategies like MO-CMA-ES [185], genetic algorithms like NSGA-II [37], differential evolution like MOEA/D-DE [103], ant colony optimization like MO-ACO [109], or particle swarm optimization like MOPSO [23]. Algorithm 1.1 shows the major steps of a generic elitist MOEA. Most of the algorithms follow indeed a similar skeleton, but differ in the specific sub-functions. Despite the richness of algorithms, NSGA-II by Deb et al. [37] remains very popular among engineers, mostly because its source code is available, and it has few parameters.

Algorithm 1.1 Generic elitist evolutionary algorithm

```
$P_0 \leftarrow$ InitializePopulation
$t \leftarrow 0$
while Stopping criterion not met do
    $M_t \leftarrow$ SelectParents($P_t$)
    $Q_t \leftarrow$ Variate($M_t$)                    ▷ Crossover & Mutations
    EvaluteFitness($Q_t$)
    $R_t \leftarrow P_t \cup Q_t$
    $P_{t+1} \leftarrow$ Select($R_t$)
    $t \leftarrow t + 1$
return $P_t$
```

Further, most available MOEAs have been developed for unconstrained optimization problems [122] and need additional constraint handling strategies (CHSs) to cope with constraints [24, 38, 58], a must for mechanical design problems. Recent results have however reported that their performance on real problems was unreliable, although the same algorithms and CHSs have good performance on typical benchmark problems of the field [59, 142]. Indeed, the latter problems have been shown to be too simple and unrealistic in general [173].

It is worthwhile mentioning that there are also deterministic multiobjective optimization algorithms. MO-DIRECT is a parameterless Lipschitzian optimizer [188, 195], but it is not adapted to constrained optimization. NOMAD is a general nonlinear optimizer based on adaptive mesh search, but it is limited to biobjective problems [6, 101]. They are thus not suitable for the applications considered in this work.

1.1.3 Automated design

Design automation is a broad field encompassing different concepts and dates back several decades—the first Design Automation Conference (DAC) from the American Society of Mechanical Engineering (ASME) took place in 1974 [126]. In this work, design automation refers to the automation of part of the design process by means of computational design synthesis (CDS), and it is seen as a superset of design optimization. CDS approaches support engineers by generating and evaluating numerous alternatives early in the design stages [15], see Figure 1.3. Their goal is to speed up and formalize steps of the process with the intent to increase the knowledge about a design task by easing feedback loops, and to allow clients to better participate throughout the process. The desired side effect is to free

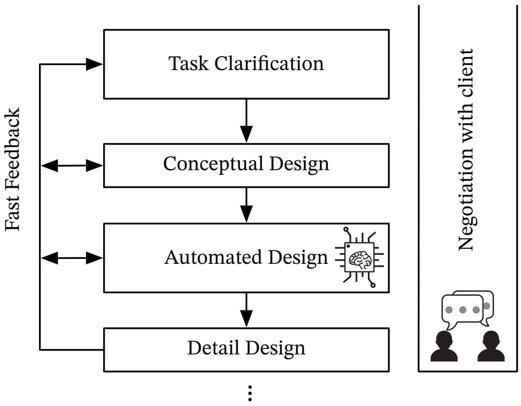


Figure 1.3: Proposed updated design process to include automated design steps.

engineers from tedious and repetitive tasks to allow them to focus on creative tasks [19].

With the advances in computers, it is nowadays even possible to consider design automation past routine tasks and to have it tackle the creative process of generating concepts [69]. Recently, researchers have been able to obtain creative behavior using generative adversarial networks (GANs), allowing machines to explore beyond known designs [21, 22]. Combined with rapid manufacturing, such tools could help drastically reduce the time needed from the definition of a design problem to the first touchable prototypes.

While research is making steady progress, the simplest form of automated design are barely used in industry as reported by several authors in 2016 [190], 2018 [153], and 2020 [154]. The technology transfer to industry is hampered by (i) a lack of awareness of existing methods and challenges to identify potential tasks [153], (ii) a lack of published methods and applicable guidelines [190], (iii) limited knowledge about the currently applied design process [154], and (iv) general mistrust in automated processes [190]. The benefits, however, are important, especially for companies with established technologies, mostly doing design variants based on updated specifications or requirements from new customers. Tailored tools fitted to their application could help boost initial design phases and cut down the time-to-market, a powerful economic argument. To promote such practices, Rigger et al. [154] proposed a methodology to conduct workshops and training sessions to guide engineering teams to identify opportunities and implement automated design approaches.

1.1.4 Engineering education

Both the lack of transfer to industry and the fear of engineers to be replaced by automation is an important motivation to look at how the education of future engineers can be adapted. In addition to increasing awareness about existing design automation methods, the skill set of future engineers should be reviewed. Indeed, Frey and Osborne [57] estimated the likelihood for jobs to get computerized and linked it to the required skills. They found that the more the jobs required skills such as “social perceptiveness”, “negotiation” or “originality” the less susceptible to computerization they were.

Many engineering accreditation bodies have already recognized the need to include “professional skills” in engineering education programs [162, 191]. Although the many names referring to “professional skills”—like “21st Century” or “soft” skills—suggest there is no unique set of skills under these labels, there is a consensus on the need for engineering education to address domains beyond strictly engineering and science, including elements of ethics, social responsibility and organizational aspects [96, 191]. Such skills also appear prominently as needs for engineers in alumni and employer surveys [16, 29, 86].

As a consequence, many institutions have done significant efforts either to amend their curriculum and introduce specialized courses [123] or to redesign their curricula by integrating interdisciplinary team-based cornerstone and capstone projects [65, 77, 179], or by following new approaches like the problem-based learning (PBL) or conceive-design-implement-operate (CDIO) approaches [30, 46]. In all cases, the focus is set on active learning mainly through projects. Multiple studies suggest that, despite these measures, the professional skill sets of graduated engineers do still not match the expectations of employers [29, 86, 96, 147].

In addition to professional skills for interpersonal interactions, engineers of all fields also need to prepare for increased interaction with machines and for this, they need “computational thinking” skills, a concept first formalized in 2006 by Wing [192]. It is defined as the thought process that allows framing a problem such that it can be carried out by a computer [66]. It puts the focus on the formulation of the problem as an important step [66] and is closely related to problem-solving. Indeed, proper framing of design requirements has been shown as an important factor on the outcome of the design process [18], even more so, if machines perform part of the task. Similarly to problem-solving skills that are best taught within disciplinary practice [191], “computational thinking” should become an important part of engineering education and the best approach to support its learning should be investigated.

1.2 Electro-mechanical actuators

In order to address the various topics and issues related to automated design, the design of electro-mechanical geared actuators will be the particular system considered, forming the backbone of this work. Electro-mechanical actuators are systems used to control the position of other components—e.g., valves, printer heads or gates. They are built around a similar set of components—a motor, a gearbox, and a housing. They can nonetheless be very different depending on their applications and requirements. Some examples are shown in Figure 1.4.

The specific design task is twofold: (i) find the components—type of motor and gearbox

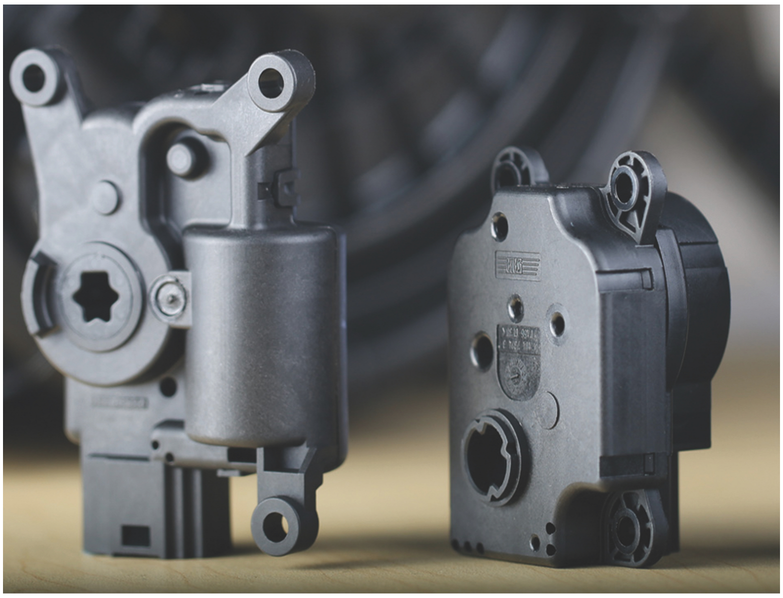


Figure 1.4: Examples of automotive electro-mechanical actuators—courtesy of Johnson Electric (JE).

technology—sometimes called configuration problem, and (ii) find their parameters that satisfy a given performance at the output shaft for different specified operating points, while ensuring mechanical integrity and packaging limitations. Actuators offer thus several research opportunities:

1. The problem is generic enough to be applicable to many real applications.

2. They are relatively simple, easing their analysis. Yet, it is a coupled multi-disciplinary system with the challenges associated to it.

3. Specifications are mostly set at the system level.

Most studies related to such systems have so far only focused on particular elements: the gearbox [40, 67, 97, 106, 169], the electric motor [150] or the packaging of components [14, 64, 170]. There are, to the best of the author's knowledge, no published work integrating all three aspects. While it is sometimes needed to develop new motor architectures, or to investigate specific gears, most actuator manufacturers rarely start from scratch, but rather look for combinations of existing motors and adapt prior gearboxes to match new or updated specifications. In this context, typical specifications only define the output performance and a packaging form factor, which are both system-level constraints. Since motor, gearbox and housing are all interdependent, the use of an integrated approach is important to suggest qualitative and ready-to-use solutions.

1.3 Problem statement

In summary, the following shortcomings have been identified:

1. Actuator design is a complex task, for which automated and integrated approaches are needed;
2. State-of-the-art MOO algorithms struggle with real-world problems;
3. Benchmark problems used to evaluate the performance of MOEAs are not realistic and representative of constrained design problems;
4. Automated design approaches are rarely used in industry;
5. Too little is known about the impact of projects on the learning of professional skills;
6. Students are underprepared for the upcoming challenges of AI-based design.

1.4 Goals and objectives

The goal of this thesis is to investigate three facets of design automation: modelling, optimization, and education, and for each to discuss the challenges and develop new tools and methodologies to address them. These topics are investigated using the design of geared electro-mechanical actuators as a common thread. Whenever possible, generic and extendable approaches are preferred. The following objectives are set to attain this goal:

1. Development of an integrated numerical model to predict the performance of many geared electro-mechanical actuator configurations suitable for different applications;
2. Implementation of an automated design tool based on multiobjective optimization using the integrated model;
3. Validation of the usefulness of the tool for real applications;
4. Numerical assessment of realistic multiobjective problems and comparison to existing benchmark;
5. Formulation of a novel and more efficient constraint handling strategy for multiobjective optimization;
6. Assessment of means to promote stronger professional skill learning of students;
7. Assessment of the behavior of students when faced with a novel automated design tool.

1.5 Outline of the thesis

The main body of this thesis is divided in four chapters.

Chapter 2—Automated Design of Electro-mechanical Actuators introduces the developed integrated model for the design of geared electro-mechanical actuators. The general architecture to build an integrated model from modular components is presented. The chapter shows how computer graphics techniques can be applied to consider system-level constraints such as packaging or ease of assembly. The obtained model is then coupled to a multiobjective optimizer and to an interactive result visualization tool to form an automated design framework. The potential of the latter is investigated through several case studies.

Chapter 3—Realistic Benchmark Problems details how a subset of the previously presented model is used to derive MODAct: a novel constrained multiobjective optimization test suite focused on constraints. Further, the chapter introduces existing and new tools to quantify the constrained search space even for not explicitly formulated problems such as many “real-world” problems. The tools are used to compare MODAct to constrained multiobjective optimization problems (CMOPs) from literature. A convergence study on the same problems using NSGA-II, NSGA-III, and C-TAEA highlights further differences to published CMOPs. The chapter emphasizes the key role constraints and in particular the number of constraints simultaneously violated play in MODAct.

Chapter 4—An Improved Constraint Handling Strategy: cEpsilon investigates the effectiveness of several published CHSs. Based on observations from the number of simultaneously violated constraints of MODAct problems, a novel per-constraint ϵ-constrained method, called cEpsilon, is introduced and evaluated. An automatic algorithm configuration approach is applied to generate several configuration variants of the considered CHSs. In total, 12 configurations are tested using 64 problems. The analysis suggests the new cEpsilon CHS to be the most competitive technique for different test suites.

Chapter 5—Educating Future Engineers presents two studies conducted with students to (i) investigate their learning of professional skills through two different kinds of projects and (ii) understand the impact on design of the use of machine intelligence. Using a questionnaire in a pre-post design, the changes in the students’ self-efficacy beliefs regarding professional skills are evaluated. The results obtained from Bachelor students having worked on an in-course project are compared to the ones from Master students having worked on a capstone project. The comparison allows identifying differences between the two formats. The effect of machine intelligence is studied by asking students to perform a design task once using a traditional approach and a second time using an automated design tool. A special online modelling platform is developed and used by students to access the component and the integrated models, as well as an optimizer. An analysis of the suggested actuators and the requests in the online platform highlights important differences in the products and in the design process.

Zu Hilfe! Zu Hilfe! Sonst bin ich verloren

Tamino (Die Zauberflöte, W.A. Mozart)

2

Automated Design of Electro-mechanical Actuators

ELECTRO-MECHANICAL ACTUATORS are small systems used in various applications where position control or rotational motion is required. They have the advantage to be relatively simple systems. Yet, they are multi-disciplinary coupled systems that require iterative work to be designed, even for experienced engineers. Indeed, actuators can be composed of different technologies and components, that need to be selected, optimized, and positioned to form a functional system, which also has to satisfy several constraints, see Figure 2.1. Design automation offers the promise to design an electric machine and a gearbox, and to package all the components in a single step.

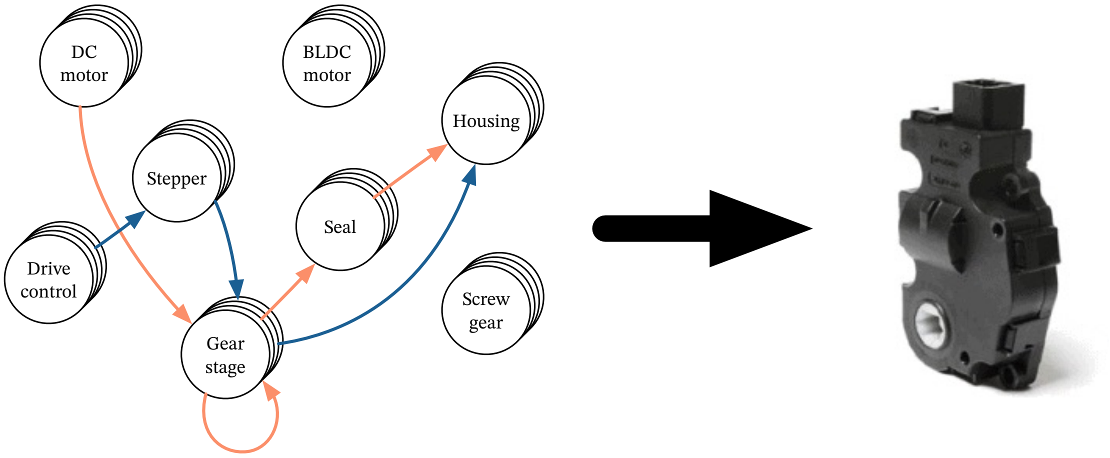


Figure 2.1: Multiple ways to make one actuator, actuator image courtesy of JE.

Various pieces of research have tackled the components of this problem: electric motor optimization [150], gearbox design [40, 67, 97, 106, 169] or packaging [14, 64, 170]. The latter, in particular, is a difficult problem both for computers and human engineers [14].

Modern computer graphics technologies, such as ray tracing, have enabled efficient methods to be developed [64]. Stochastic meta-heuristics are the optimization method of choice in all of these studies.

This chapter exposes how all the components are assembled to build an integrated model of an actuator that performs component-specific simulations and a system-level analysis. More than component modelling accuracy, the focus is on modelling system-level constraints, since those are the requirements defined by the customers. Through the use of 3-D meshes to represent the components within a whole system, packaging or ease of automated assembly can be considered in the model. Following the modelling details, several case studies are presented to highlight the potential and advantages.

Part of the content of this chapter has been published in:

C. Picard and J. Schiffmann. “Automated Design Tool for Automotive Control Actuators”. In: *IDETC-CIE2020*. ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 11B: 46th Design Automation Conference (DAC), Aug. 17, 2020. DOI: 10.1115/DETC2020-22390

C. Picard and J. Schiffmann. “Realistic Constrained Multiobjective Optimization Benchmark Problems From Design”. In: *IEEE Transactions on Evolutionary Computation* 25.2 (Apr. 2021), pp. 234–246. DOI: 10.1109/TEVC.2020.3020046

Author contributions: CP and JS designed research; CP performed research, and wrote the articles himself.

2.1 Building an integrated model

For the purpose of this work, an actuator is considered as a sequence of $k+1$ components inside a housing. A motor first transforms electrical power into mechanical power that then gets conditioned by successive stages of gearbox technologies and shafts. This power chain also forms the backbone of the layout, since the position of the components exchanging power is constrained. With this in mind, it is possible to build an integrated model that remains modular: component-specific discipline is hidden behind a generic component interface upon which system-level models are added. This approach is similar to the object-oriented paradigm in computer science.

To be interoperable, each component i needs:

1. a cost model c_i;
2. a physical model for predicting the output speed ω_i and torque T_i depending on input conditions, and component-specific constraints $\boldsymbol{g}_i$;

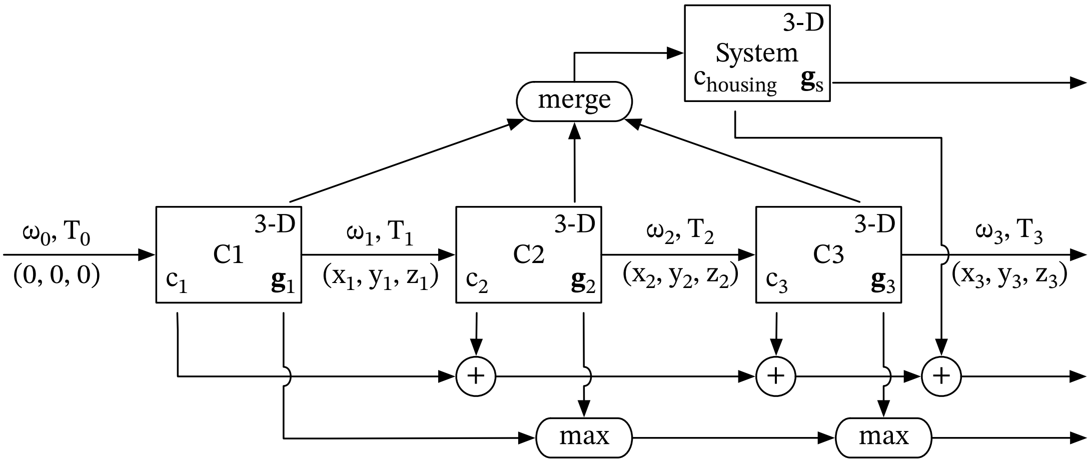


Figure 2.2: Schematic representation of the way individual components are connected to form an integrated model.

3. a 3-D tessellated description or 3-D triangular mesh, including spatial configuration rules giving the coordinates (x_i, y_i, z_i) for the next component.

Figure 2.2 illustrates the architecture that allows the integration of these components into a system. Starting from a driving condition for the first component (ω_0, T_0), and an initial reference coordinate, the system is iteratively built by passing the output condition and position to the next components. The matching component-specific constraints are aggregated, keeping the worst case. All 3-D meshes are merged to form a 3-D representation of the whole system, which is used to predict the volume—bounding box or convex hull—of the whole assembly and upon which additional queries can be performed. The following sections describe the component model details and how they are combined to form a system.

This modular approach offers great flexibility. New components can be added whenever needed and as long as the interface is unaffected, the inside model of each component can be changed. So, improved accuracy or additional physics can be added incrementally. As such, it forms an actuator modelling framework.

The framework itself is coded in Python [146], an open-source programming language that offers a strong set of open-source libraries and that allows an easy interoperability with libraries from other languages. Indeed, many parts of the algorithms presented and used in this work take advantage of fast C or C++ routines.

2.2 Component models

For all components, fast models with acceptable accuracy have been preferred over detailed and expensive models, since the advantage of speed seems particularly important for tools used in the early design phases. The following component models have been implemented:

(i) stepper motors, (ii) DC motors, (iii) spur gears, (iv) crossed helical gears, and (v) planetary gears. The stepper motor and spur gear models have been written by the author and are described in the following subsections, while the DC motor, crossed helical gear and planetary gear models have been coded by Spaeth [165] for his Master thesis under the author's supervision.

2.2.1 Stepper motor

Physical model

Stepper motors are permanent magnet motors that are most often controlled in an open-loop manner by switching the stator phases to trigger steps. Since the control is open-loop, the absolute position of the rotor is in theory unknown. However, in normal operating conditions, it is assumed that no steps are lost, i.e., every electrical step results in a mechanical step. In that case, the rotational speed of the rotor ω_m is related to the electrical stepping pulsation ω_{el} through the number of rotor teeth p[1]: $\omega_{el} = p\omega_m$.

In addition to a step-by-step control, stepper motors can also be used in a continuous operation mode, where the steps are performed continuously. This is called high-speed operation [1] and resembles the operation of brushless DC (BLDC) motors, except the steps are executed with no feedback.

In general, the simulation of the dynamic behavior of stepper motors is done by solving the coupled ordinary differential equations (ODEs) of the electrical circuit and the mechanical motion [157]. The prediction of the speed-torque characteristic of steppers is obtained by repeatedly solving these ODEs for various speeds and loads until step losses are detected. Even though an optimized C++ implementation by the project's industry partner exists, the procedure to get the full speed-torque characteristics for a stepper takes several seconds. This is impractical for integrated modelling and optimization purposes.

Instead, assuming high-speed and steady operations, an algebraic torque equation can be derived based on published recommendations by Acarnley [1, Eq. (5.14)].

$$T_{po} = \frac{k_m V}{\left(R^2 + \omega_{el}^2 L^2\right)^{1/2}} - \frac{\omega_m k_m^2 R}{\left(R^2 + \omega_{el}^2 L^2\right)} \tag{2.1}$$

with $V = 4V_s/\pi$ for a two-phase-on scheme with the supply voltage V_s, k_m the motor constant, R the total circuit resistance, and L the winding inductance.

The original equation (2.1) misses important physical considerations present in the ODEs of Johnson Electric (JE), such as magnetic flux saturation in the stator, maximum current

---

[1] p can thus be considered as a "transmission ratio" since it links to electric input "speed" to the mechanical output speed.

limitation, friction, and temperature-dependent flux and resistance. It is, however, possible to include these concepts in this steady-state equation.

Induction machines are often simulated with the hypothesis of linear magnetics, i.e., the magnetic flux is proportional to the current applied in the windings. For small steppers or stepper stators made out of less premium steel, however, the flux path can become saturated, reducing the useful flux and thus the torque at the rotor side. In the ODEs, saturation is accounted for by penalizing the motor constant for high currents.

$$k_m = k_{m0}\mu(N_w I) \tag{2.2}$$

$$\text{with } \mu(x) = \mathrm{e}^{-\left(\frac{x}{NI_{sat}}\right)^2} \tag{2.3}$$

where k_{m0} is the unsaturated motor constant, N_w is the number of turns in the windings, I is the instantaneous current through the windings, and NI_{sat} is the saturation constant.

To be included, some steady-state equivalent current needs to be derived. Eq. (2.1) can be transformed by factorizing k_m into Eq. (2.4) and since the units of k_m are $\mathrm{N\,m\,A^{-1}}$, the content of the parenthesis can thus be interpreted as an equivalent current.

$$T_{po} = k_m \left(\underbrace{\frac{V}{\left(R^2 + \omega_{el}^2 L^2\right)^{1/2}}}_{I_s} - \underbrace{\frac{\omega_m k_m R}{\left(R^2 + \omega_{el}^2 L^2\right)}}_{I_{em}} \right) \tag{2.4}$$

The obtained $I_{eq} = I_s - I_{em}$ is an average current, while Eq. (2.2) uses an instantaneous current. To bridge the two, a constant correction is assumed $I = k_s \cdot I_{eq}$. The effect of saturation can then be calculated with I_{eq} in Eq. (2.2). Since I_{eq} depends on k_m, loops are needed to converge on both values. However, for the sake of speed, this step is omitted, and instead, has been factored into k_s, whose value is obtained through tuning ($k_s = 1.15$).

In the physical system, current is limited by the drive controlling the stepper. The limitation is implemented by reducing the apparent voltage seen by the windings by means of pulse-width modulation (PWM). Following a similar approach, the supply voltage is reduced to respect a maximum current $I_{\max}$ when set.

$$V_s' = \min\left\{ \frac{V_s}{k_s} \,;\, (I_{\max} + I_{em})\left(R^2 + \omega_{el}^2 L^2\right)^{1/2} \frac{\pi}{4} \right\} \tag{2.5}$$

Finally, the viscous $\omega_m f_{dyn}$ and Coulomb T_c friction torques are added to the model, and the effects of a difference of ambient temperature are built into the motor parameters R and k_{m0}. Given a set of motor parameters obtained for example through finite-element method (FEM) simulations, and winding parameters N_w and d_w, the diameter of the coil wire, the available torque T_m is calculated using Eq. (2.6) to (2.12).

$$
\begin{aligned}
f_{el} &= 1 + \alpha_{el}(T_s - T_{nom}) && (2.6)\\
f_{mag} &= 1 - \alpha_{mag}(T_s - T_{nom}) && (2.7)\\
R &= R_{nom}\frac{N_w d_{w,nom}^2}{N_{w,nom} d_w^2} \cdot f_{el} && (2.8)\\
k_{m0} &= p N_w \phi_0 \cdot f_{mag} && (2.9)\\
L &= L_{nom} {N_w}^2 && (2.10)\\
I_{eq} &= \frac{4V_s'/\pi}{\left(R^2 + \omega_{el}^2 L^2\right)^{1/2}} - \frac{\omega_m k_{m0} R}{\left(R^2 + \omega_{el}^2 L^2\right)} && (2.11)\\
T_m &= \max\left\{k_{m0}\mu\left(N_w I_{eq} k_s\right) I_{eq} - \omega_m f_{dyn} - T_c \,;\, 0\right\} && (2.12)
\end{aligned}
$$

where α_{el} and α_{mag} are material properties, T_s is the actual system temperature, T_{nom}, R_{nom}, $d_{w,nom}$, $N_{w,nom}$ and L_{nom} are the nominal motor parameters, and ϕ_0 the linkage flux.

In this formulation, the possible values for N_w and d_w are motor dependent. In order to allow for a dimensionless control over the windings, two scaling parameters are introduced: the fill factor FF and the resistance scaling factor R_{scale}, so that, the windings of a base motor can be adapted and its performance simulated.

$$
\left.\begin{aligned}
FF &:= \frac{N_w d_w^2}{N_{w,nom} d_{w,nom}^2}\\[2ex]
R_{scale} &:= \frac{N_w d_{w,nom}^2}{N_{w,nom} d_w^2}
\end{aligned}\right\}
\Leftrightarrow
\left\{\begin{aligned}
N_w &= N_{w,nom}\sqrt{FF}\\[2ex]
d_w &= \sqrt[4]{\frac{FF}{R_{scale}}}\, d_{w,nom}
\end{aligned}\right.
\qquad (2.13)
$$

With this, the simplified algebraic model to predict the speed-torque characteristic of stepper motors is complete. Figure 2.3 illustrates the good agreement between this new steady-state model and the ODEs. It also shows the important effect flux saturation and current limitation have on the torque at low speed, and that their additions can capture the same effects. Even if the original Eq. (2.1) is applicable only to high-speed operations, the additional model features allow the model to be used for predicting low-speed torques. Further comparisons between the two models on different steppers are shown in Appendix A.

With a steady-state model between 10 000 and 100 000 times faster than the resolution of the ODEs, the loss in accuracy is largely compensated. In the end, it remains the users' choice to decide if the accuracy is sufficient for their needs. In particular, there are several tunable parameters that can be further adjusted, based on experimental data for example.

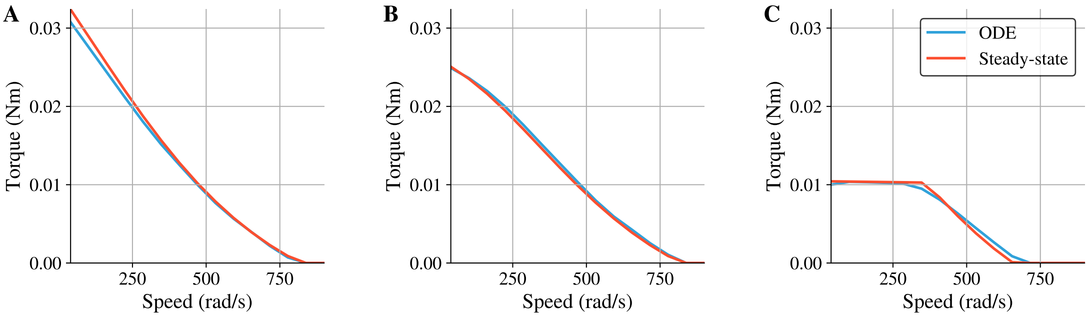


Figure 2.3: Comparison between the speed-torque characteristics obtained through the simulation of the ODEs and through the steady-state model, highlighting the effect of different features: (A) stepper with friction, (B) adding saturation and (C) adding a current limitation.

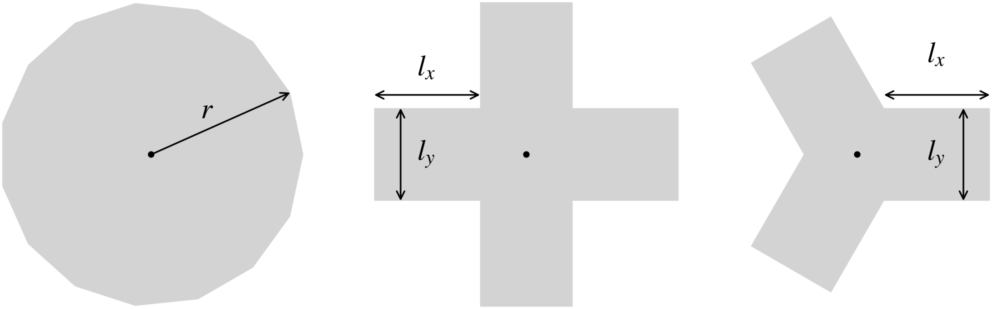


Figure 2.4: Representation of the available mesh geometries for the motors and their parameters.

Cost model

The cost of a motor c_m is calculated based on the volume and materials for the rotor, the stator and the windings, assuming a density ρ and a specific cost per kilo $\bar{c}$ for each material. Since the rotor and stator geometry are fixed for a given stepper motor, the cost can be split into a fixed contribution and a variable contribution depending on the fill factor. The cost is thus linear with respect to FF.

$$c_m = c_{\text{fixed}} + c_{\text{variable}} \tag{2.14}$$

$$c_{\text{fixed}} = V_{\text{rotor}}\rho_{\text{rotor}}\bar{c}_{\text{rotor}} + V_{\text{stator}}\rho_{\text{stator}}\bar{c}_{\text{stator}} \tag{2.15}$$

$$c_{\text{variable}} = FF\, V_{\text{Cu,nom}}\rho_{\text{Cu}}\bar{c}_{\text{Cu}} \tag{2.16}$$

3-D model

Three different stepper body geometries are available, see Figure 2.4: a cross, a 3-branch star and a cylinder. The dimensions are obtained from the corresponding motors.

2.2.2 Spur gears

Spur gears are extremely common in gearboxes and their modelling is well-documented and governed by numerous norms [79–82, 183, 184]. The implemented model follows the ISO norms [79–82] with a few exceptions that are highlighted below.

Each gear is defined by a number of teeth Z, a profile shift coefficient x, a module m, a thickness b, a tooth profile h^*_{fp}, ρ^*_{fp}, and h^*_{ap}, a pressure angle α, and a material, see ISO 1122-1:1998 [82] for a definition of the terms and ISO 21771:2007 [81] for the geometrical relations. A gear stage is composed of two gears: the pinion (subscript 1) and the wheel (subscript 2). Their numbers of teeth fix the input-output speed relation $i = Z_2/Z_1$—also called transmission ratio. To be able to engage properly, two gears need to share the same module and to respect four geometrical constraints:

1. avoid tooth interference along the contact path;
2. have a sufficient contact ratio for smooth operation ($\epsilon_\gamma \geq 1.1$);
3. limit large specific sliding at the engagement point ($\zeta_{f1} \geq -5$), and
4. at the disengagement point ($\zeta_{f2} \geq -5$).

Beyond these geometrical considerations, the chosen materials and their properties need to be included. Since, in the targeted applications, the gears are manufactured by polymer injection, there are some differences to standard metal gears [111]:

1. any module value can be selected;
2. the efficiency of the transmission is set to a fixed 95% since it is hard to predict and depends on the applied lubricant, the type and density of fiber reinforcement, the humidity levels and wear [125];
3. the tooth root σ_F and tooth flank σ_H stresses are calculated following the ISO 6336:2006 method B [79] and VDI norms [183, 184], with $K_{H\beta} = K_{H\alpha} = K_{F\beta} = K_{F\alpha} = 1$ and $K_A = 1.25$;
4. the calculation of the mechanical safety factors and fatigue damage fraction is adapted to include temperature-dependent fatigue based on Wöhler curves, also called S-N curves [160], provided by the polymer manufacturers.

To maintain their mechanical integrity over their entire lifetime, the safety factors need to be ≥ 1 and the damage fraction ≤ 1.

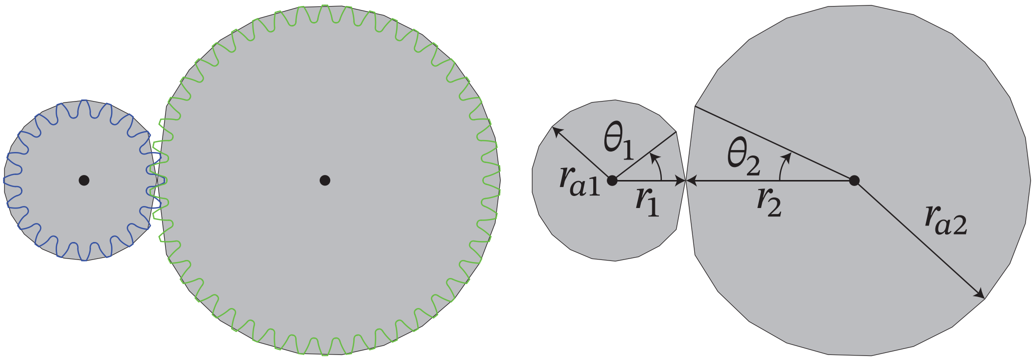


Figure 2.5: Planar view of the special geometry used to model a gear pair: with respect to the real gear geometry (left) or with the key variables (right).

Cost model

The cost of a gear pair is based on the approximate cylindrical volume of the pinion and the wheel, assuming a density ρ and a specific cost per kilo $\bar{c}$ for each material.

$$c_{gp} = \frac{\pi}{4}(d_1^2 b_1 \rho_{\text{mat1}} \bar{c}_{\text{mat1}} + d_2^2 b_2 \rho_{\text{mat2}} \bar{c}_{\text{mat2}}) \tag{2.17}$$

3-D model

The 3-D representation of gears requires further considerations. On the one hand, generating the exact toothed geometry is certainly overly complex. On the other hand, gears are often represented as simple cylinders of the primitive diameter $d = mZ$ of the gear, which underestimates their volume, an important point for the system-level modelling. It is also possible to use cylinders with the addendum circle diameter d_a, but then, the meshes of a gear pair are overlapping in the contact zone. In order to combine both advantages, a “cut” cylinder geometry is created, as shown in Figure 2.5. It is a cylinder with a diameter of d_a, except at the point of contact between the pinion and the wheel, where it is truncated to d. The cut angle θ is set to $\theta = \cos^{-1}(d_f/d_a)$, where d_f is the root circle diameter.

2.3 System-level models

Now that the components are defined, they can be assembled to form a system. Although the code is more generic, the explanations are, for the sake of clarity, restricted to systems composed of a motor followed by k gears. Some quantities can be calculated directly, like the geometrical gear constraints, or by aggregating the individual contributions, such as the cost of the components c_{comp}, Eq. (2.18), the gearbox transmission ratio i_{gb}, Eq. (2.19), or the total transmission ration i_{tot}, Eq. (2.20).

$$c_{\text{comp}} = \sum_{i=0}^{k} c_i \tag{2.18}$$

$$i_{gb} = \prod_{i=1}^{k} \frac{Z_{i,2}}{Z_{i,1}} \tag{2.19}$$

$$i_{tot} = p \cdot \prod_{i=1}^{k} \frac{Z_{i,2}}{Z_{i,1}} \tag{2.20}$$

Operating condition dependent constraints—the stresses in the gears—and system-level quantities can only be calculated once the state of the system is known. This means:

1. calculating the operating conditions of all components;
2. positioning the components with respect to each other.

The following subsections first introduce the procedure applied to the operating conditions. Second, the packaging operation is described, followed by several algorithms performing queries on the obtained 3-D model of the system.

2.3.1 Operating conditions

An operating condition is defined as a set of supply voltage, a maximum current, a system temperature, a rotational angle or speed, a torque, the number of cycles to consider for this point and its mode. The system considers two types of operating mode:

1. In *running* mode, the output rotates continuously at a given speed.
2. In *holding* mode, the output rotates by a small angle.

The operating points for all components are resolved sequentially by propagating the output condition of a component to the next. The procedure requires the driving conditions—i.e., the input conditions to the first component—to be given. They can be imposed by the users, or calculated to match desired output speed or torque. The most common and straightforward approach is the speed-matched approach, where the driving speed is set from the desired output speed using the total transmission ratio: $\omega_{in} = \omega_{out} \cdot i_{tot}$. The same approach can be applied to the output angle in the holding mode.

The holding mode is similar to the running mode, except for the handling of the number of cycles. In running mode, the number of cycles performed by each component is directly related to its rotational speed. Whereas in a holding situation, since the components are stationary, all components are loaded and unloaded once per cycle. In practice however, for actuators with large gearboxes, a very small rotation of the output can still represent several rotations at the motor, and thus also several loading cycles. For this reason, in holding mode, an angle information is passed from component to component, and the number of cycles gets adapted in consequence.

Once the operating conditions are resolved for all components, it is possible to calculate the effective stresses in the gears. When a set of operating points is defined, the procedure is repeated for each point. The resulting safety factors are aggregated, and only the smallest for each component is kept. For the fatigue, the damage fractions from each condition are summed per component to form the cumulative damage fraction. The constraints can be further reduced, by keeping only the worst value over components of the same kind. For example, the minimum safety factor for the tooth root stress is given by:

$$S_{F,\mathrm{min}} = \min\left\{S_{F,1}, \dots, S_{F,k}\right\} \tag{2.21}$$

It is also possible to aggregate all safety factors using the harmonic mean:

$$S_{\mathrm{harm}} = \left(\frac{\sum_{i=1}^{k} S_{F,i}^{-1} + S_{H,i}^{-1}}{2k}\right)^{-1} \tag{2.22}$$

It is important to note that the stresses, and hence safety factors, are calculated based on the effective loads acting on the gears, that might not match with the desired load. The difference between the desired and effective torque at the output is called the torque excess, defined for the j^{th} operating point as:

$$\Delta T_j = T_j - T_{\mathrm{desired},j} \tag{2.23}$$

The advantage of this approach is that it allows evaluating the resulting trade-offs of more powerful actuators. When this is not desired, the code offers a “limit torque” feature, which, if the system has some torque excess, loops once more to recalculate the loads for each component to match the desired output torques.

At this stage, other quantities related to the operation of the actuator can be calculated, such as for example, the electrical to mechanical energy conversion efficiency η_j of the j^{th} operating point:

$$\eta_j = \frac{\omega_j T_j}{V_s I_{eq,j}} \tag{2.24}$$

2.3.2 Packaging

Packaging consists in defining the coordinates of all the components. In general, each component being a rigid body, its position in a given reference frame is given by six variables: three coordinates (x, y, z) and three angles (ϕ, θ, ψ). In an actuator however, there are fewer degrees of freedom and successive components are positioned relative to the previous one

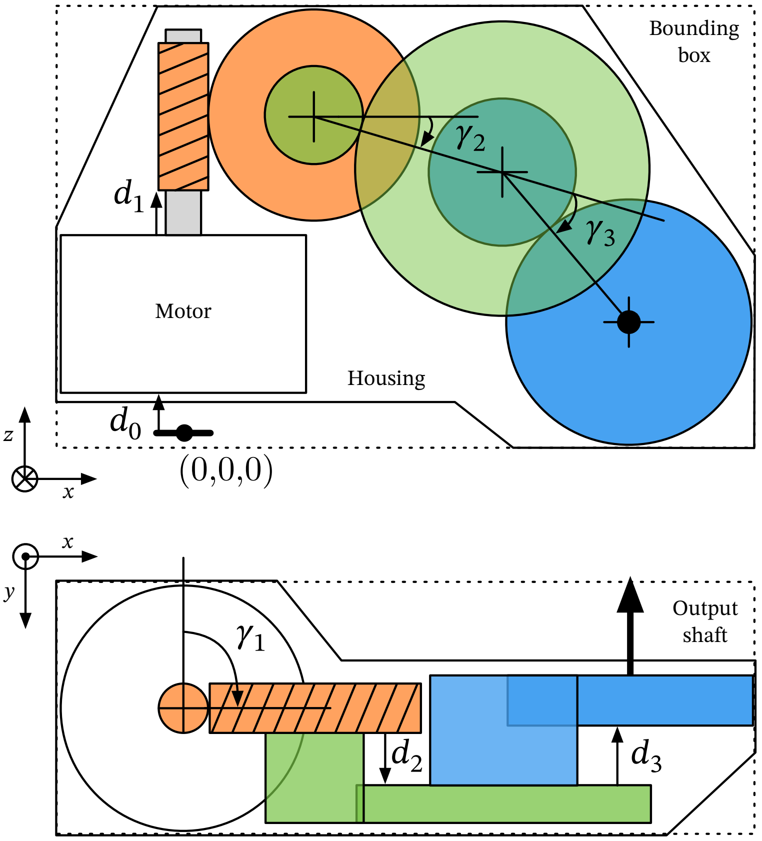


Figure 2.6: Schematic representation of an actuator with a crossed helical gear stage and two spur gear stages, showing the relative positioning procedure of the components, and the resulting bounding box (side and top views).

using only two variables: (i) a translation d_i along the rotational axis (the shaft), and (ii) a rotation γ_i around that same axis, see Figure 2.6. Within components, additional translations or rotations are possible, e.g., in crossed helical gears, but their definition is controlled within the components.

The absolute positions are calculated sequentially by starting from a reference frame arbitrarily set at $O = (0,0,0)$. There are two procedures depending on whether the position of the motor shaft or the output shaft should be known *a priori*:

1. The *forward* mode calculates the coordinates from the motor to the output—as shown in Figure 2.6;
2. The *backward* mode starts from the output to the motor.

The coordinates from a component used as a starting point for the next translation and rotation is called a *hook*. Each component defines a set of hooks for both procedures, and for positive and negative values of d_i. Figure 2.7 shows the positions of the hooks for the different cases for spur and crossed helical gears. Figure 2.8 compares the effects of the

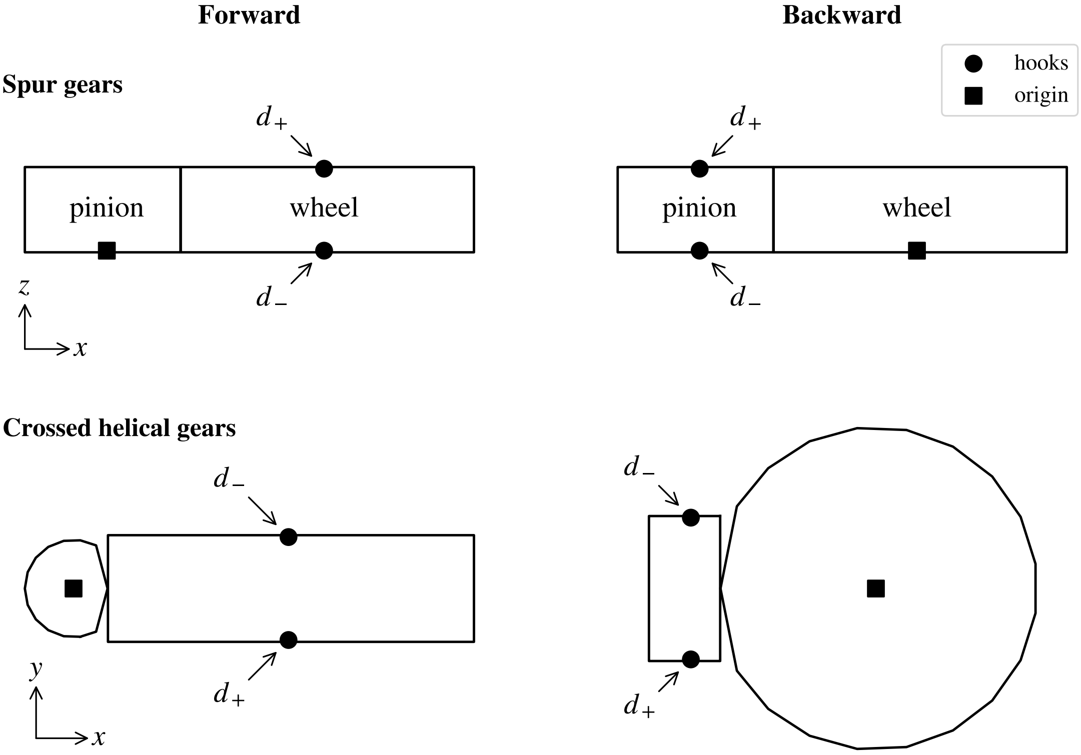


Figure 2.7: Location of the positive and negative hooks for spur gears and crossed helical gears in forward (left) and backward (right) mode

two procedures on the resulting layout. The translation is implemented in the model by "stretching" the gears, as in Figure 2.6, or by adding a shaft, as in Figure 2.8. In both cases, the additional element is assumed infinitely rigid.

Once the coordinates of all components are known, the 3-D mesh of each component can be generated and assembled to form a mesh of the whole system. The meshes are managed through the Trimesh [34] library. With this system mesh, it is then possible to estimate various volumes, such as the bounding box—the smallest axis-aligned box—or the convex hull—the smallest convex polyhedron. The latter is also used to predict the housing shape. Similarly to other components, the cost of the housing is volume-dependent and is calculated based on its surface A_h and on the chosen material, assuming a fixed thickness h. Adding it to the cost of the components yields the estimated total cost of the system.

$$c_{\text{housing}} = h A_h \rho_h \bar{c}_h \tag{2.25}$$

$$c_{\text{tot}} = c_{\text{comp}} + c_{\text{housing}} \tag{2.26}$$

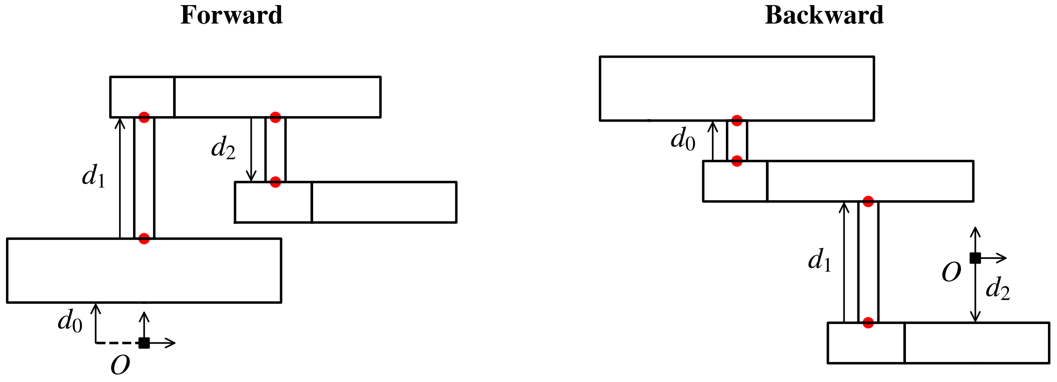


Figure 2.8: Comparison of the resulting actuator layout for the forward (left) and backward (right) modes, for an actuator composed of a stepper motor and two spur gear pairs.

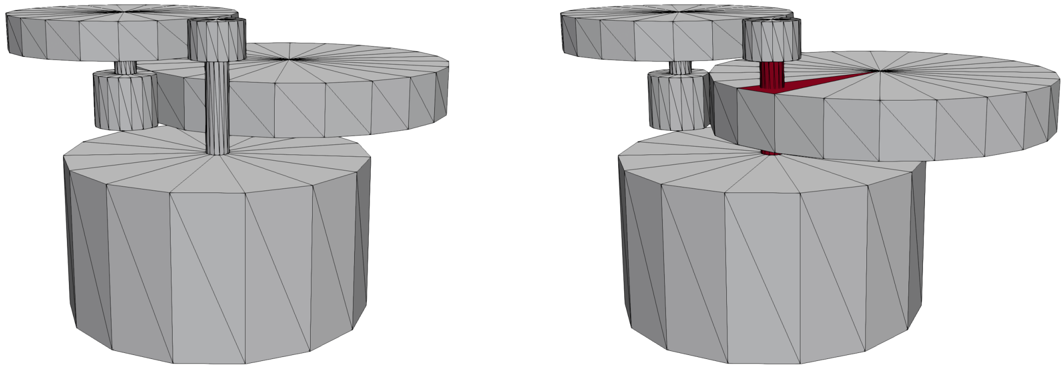


Figure 2.9: Illustration of an actuator with and without internal collision. Colliding faces detected by FCL are shown in red.

Internal collisions

Most importantly, the 3-D meshes enable valuable system-level constraints to be evaluated. One of these is the internal collision detection. Indeed, the layout procedure does not enforce that the resulting actuators are physically possible, i.e., that no components are colliding with each other, see Figure 2.9. This collision detection is performed with the C++ *flexible collision library* (FCL). FCL organizes the space in bounding volume hierarchies and applies efficient sweep-and-prune algorithms to reduce the number of required intercomponent collision checks [135]. The only requirement is that the meshes use triangular faces.

The goal is not just to have an indicator function, but to quantify an "amount of collision". The proposed approach groups the components by shaft; since by construction, elements on a shaft cannot collide and form a single part. The procedure detailed in Algorithm 2.1 calculates a score with the number of colliding faces detected by FCL, normalized by the total number of faces of the meshes. The algorithm typically requires about 100 µs. The obtained score is equal to 0 for valid layouts and increases proportionally with the number of collisions. In order to ensure sufficient vertical spacing between the components, the component meshes are generated with a changeable 0.2 mm excess thickness.

Algorithm 2.1 Assess collisions between groups of components

```
Require:
    $A = \{M_i\}_{i=1}^n$ a set of $n$ group meshes

procedure InternalCollision($A$, with_inclusion)
    $n_f \leftarrow \sum_{i=1}^n |\text{Faces}(M_i)|$
    $F_c \leftarrow$ broad-phase collision query($A$)        ▷ Face collision query with FCL
    $\delta \leftarrow |F_c|/n_f$
    return $\delta$
```

Inclusion within a given external hull

In many applications, actuators are required to have a given form factor, such as the two examples shown in Figure 2.10. These specifications are typically necessary when actuators are further integrated into larger existing systems, such as heating, ventilation, and air conditioning (HVAC) units. Modelling this constraint is thus key, since it can be very restrictive in terms of acceptable design. Its restrictive nature makes it particularly important that the numerical value of the constraint reflects how much of an actuator is outside the hull, and thus guides the optimizer towards valid solutions.

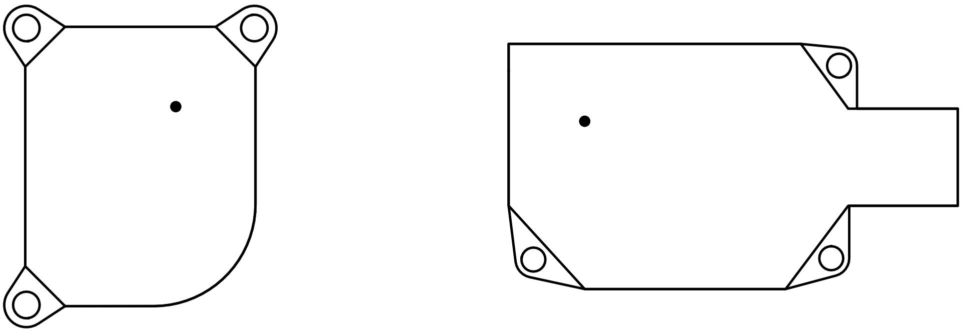


Figure 2.10: Examples of possible hulls for actuators where the black dot illustrates the position of the origin of the reference frame and corresponds to the required position of the output shaft. The volumes are obtained by extruding these shapes.

In that context, collision detection algorithms are insufficient to have an inclusion score that increases the more an actuator protrudes from the hull. Such a score can be obtained by identifying the vertices of the actuator that are outside the hull and by calculating their distance to the nearest point on the surface of the hull. The latter is done using a nearest-neighbor search method, such as the one implemented in Trimesh. It uses a k-d tree—a data structure used to partition space—to speed-up the search [118].

A naive approach would be to calculate for all vertices the signed distance to the nearest point on the hull, where the sign indicates whether the vertex is inside (positive) or outside (negative) the surface. So, the point the furthest outside the hull is the most negative one and its distance can be used as a score. In the naive approach, the computation time is constant for a given problem, and depends on the number of vertices in the actuator mesh and in the hull mesh.

Algorithm 2.2 Selective method to assess if an actuator is within a hull

```
Require:
    $A = \{M_i\}_{i=1}^{n}$ a set of $n$ group meshes
    $H$ the mesh of the hull

procedure InsideOtherMesh($A$, $H$)
    $F_c \leftarrow$ broad-phase collision query($A$, $H$)        ▷ Face collision query with FCL
    $G \leftarrow$ GroupMesh($F_c$)                              ▷ Get set of groups in collision

    $\delta \leftarrow 0$
    $V \leftarrow \{\}$                     ▷ Set of candidate vertices for distance calculation
    for all $g \in G$ do
        if Inside($g$, $H$) then                 ▷ Give a fixed penalty if $g$ just touches $H$
            $\delta \leftarrow \delta + 10^{-3}$
        else
            $V \leftarrow V \cup$ Vertices($g$)
    $O \leftarrow \{g \in A \setminus G \mid \neg$Inside(Center($g$), $H$)$\}$
    $V \leftarrow V \cup$ Vertices($O$)
    $\delta \leftarrow \delta + \frac{|\min \text{SignedDistance}(V, H)|}{\text{BoudingBoxLength}(A)}$
    return $\delta$
```

Applying the expensive nearest-neighbor search to all vertices of the actuator mesh is the bottleneck of the procedure. In particular, as actuators are closer to fulfilling the requirement, lots of time is lost calculating neighbors of points inside the hull that are of no interest for the inclusion score. So, if faster methods to filter the vertices that are outside exist, the best-case complexity of the naive method could be improved. Collision detection and ray tracing are two such methods. Ray tracing is a rendering technique commonly used in computer graphics to simulate the path of light to accurately draw textures and shadows. Ray-tracing algorithms have been extensively optimized for speed. They cast virtual rays through 3-D scenes and record the faces that have been hit. This principle can be applied to evaluate if a point is inside a mesh by casting a ray of light from it in a random direction. The point is inside if the ray hits the hull an odd number of times. The Embree [187] library handles the ray-tracing procedure itself.

The selective inclusion test procedure, detailed in Algorithm 2.2, applies a first filter through collision detection. The groups in collision with the hull are either inside but in contact with the casing or at least partially outside. In the first case, a fixed penalty is added to the inclusion score, while in the second, the vertices of the group are added to a list of vertices to test. The second filter evaluates if the remaining groups are outside or inside by applying the ray-tracing method described above to the center point of these groups. The vertices of the groups that are outside are also added to the list of vertices, which is then passed to the signed distance calculation. Finally, the distance of the point the furthest away is added to the inclusion score after normalizing the distance by the largest extent of the bounding box.

Since the procedure returns a violation distance, it is continuous and decreasing as the components are moved inside. In order to illustrate this, Figure 2.11 shows the evolution of the constraint value as an actuator fully inside a given hull is rotated away. It also highlights that the procedure can handle any hull shape.

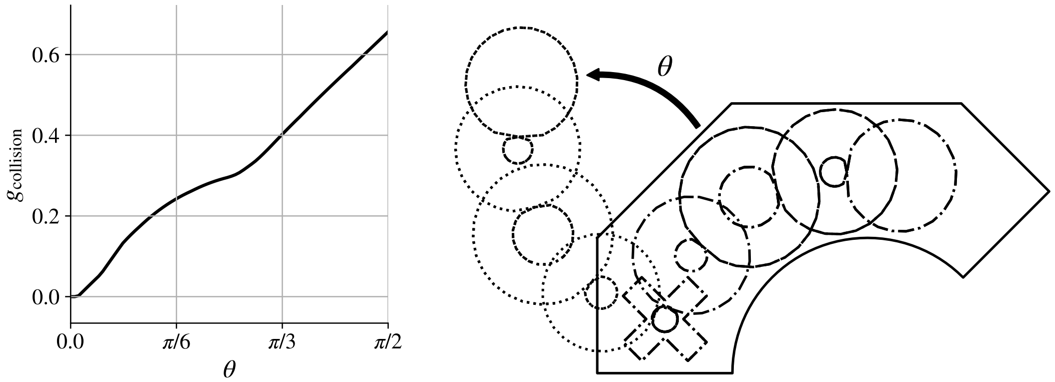


Figure 2.11: Evolution of the constraint value for the inclusion in the hull as the actuator is rotated around the first gear pair by an angle $\theta = \pi/2$.

In terms of speed, in many cases and in particular as more actuators respecting the constraints are found, it reduces the number of vertices included in the expensive distance calculation from several hundreds to tens. Indeed, running the same virtual experiment as before but focusing on execution time, see Figure 2.12, a clear speed-up can be noticed when comparing the naive and selective methods, especially for low violation cases. Yet, the computation of this constraint remains, comparatively to the previous one, much more expensive.

Many mesh formats, such as STL, can be loaded, allowing objects generated through standard computer-aided design (CAD) software to be used. Yet, to minimize the computational effort and the optimization complexity, the hull objects should be as simple and as permissive as possible. When the inclusion score is calculated, the “backward” layout mode is the most suitable and the origin of the hull should represent the desired output position.

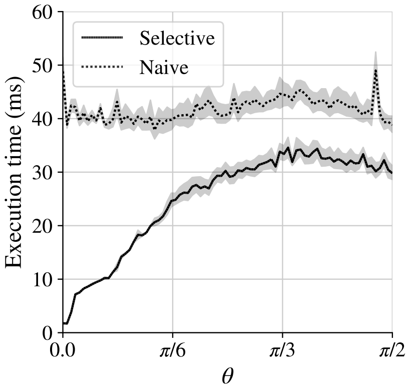


Figure 2.12: Comparison of the naive versus selective method in terms of execution speed for the example shown in Figure 2.11—calculations done on a 2.7 GHz *Quad-Core Intel Core i7* laptop, number of vertices: 96 for the hull and 252 for the actuator.

2.3.3 Assembly feasibility assessment

In many applications, actuators are mass-produced, and automated assembly lines are increasingly used to achieve high through-put and reliability. Typically, the assembly process involves inserting the components in the bottom part of the housing, that is then closed as a final step. Such process is well suited to automation, but automated assembly lines are expensive. Their cost roughly depends on the number and the complexity of each assembly step. For this reason, actuators that can be assembled in few simple steps are preferred.

Inspired by Grignon and Fadel [64], ray tracing is applied, in this context, to determine if shaft groups can be inserted following a single translation, and the minimum number of steps to perform the assembly. An assembly dependency graph is built by identifying the components that shadow each other along the assembly axis—usually aligned with the output axis. For each group, rays are cast from its vertices in the assembly direction, and collisions with faces of other groups add edges in the graph. The minimum number of assembly step corresponds to the height of the obtained graph, while deadlocks are detected by the presence of loops. The details of the procedure are summarized in Algorithm 2.3.

The advantage of being able to predict the ease of assembly are highlighted through several illustrative examples. Figure 2.13 compares two different actuator layouts for four-stage actuators: on the one side, a step-like layout, on the other, an interlaced layout. Both can be assembled with an automated assembly line, but the first needs at least five steps, while some shaft groups could be inserted simultaneously in the second, thus reducing the number of steps to two. Figure 2.14 shows that the same algorithm can be applied with components with different shaft directions. Finally, Figure 2.15 shows an example of an actuator with an assembly deadlock. Indeed, the motor and shaft group ① are both a requirement and a dependency of each other.

Algorithm 2.3 Automated assembly evaluation procedure

Require: $A = \{M_i\}_{i=1}^{n}$ a set of n group meshes and $\boldsymbol{d}$ the preferred assembly direction

```
procedure AssemblyScore($A$, $\boldsymbol{d}$)
    $\boldsymbol{d} \leftarrow$ FindAssemblyDirection($A$, $\boldsymbol{d}$)        ▷ Direction of the output shaft or $\boldsymbol{d}$
    $V \leftarrow \bigcup_{i=1}^{n}$ Vertices($M_i$)
    $E \leftarrow \{\}$        ▷ Set of dependency edges
    $F_{c+}, R_+ \leftarrow$ RayIntersectAll($A$, $V$, $\boldsymbol{d}$)
            ▷ All faces of $A$ intersected by rays from $V$ cast in direction $\boldsymbol{d}$
    for all $f, r \in F_{c+} \times R_+$ do
        $target \leftarrow$ MeshIndex(f)
        $origin \leftarrow$ MeshIndex(r)        ▷ Get mesh associated with $f$ or $r$
        if $target \neq origin$ then
            $E \leftarrow E \cup \{(target, origin)\}$
    $F_{c-}, R_- \leftarrow$ RayIntersectAll($A$, $V$, $-\boldsymbol{d}$)
    for all $f, r \in F_{c-} \times R_-$ do
        $target \leftarrow$ MeshIndex(r)
        $origin \leftarrow$ MeshIndex(f)
        if $target \neq origin$ then
            $E \leftarrow E \cup \{(target, origin)\}$
    if HasLoops($E$) then
        return $n + 1$
    else
        return Height($E$)
```

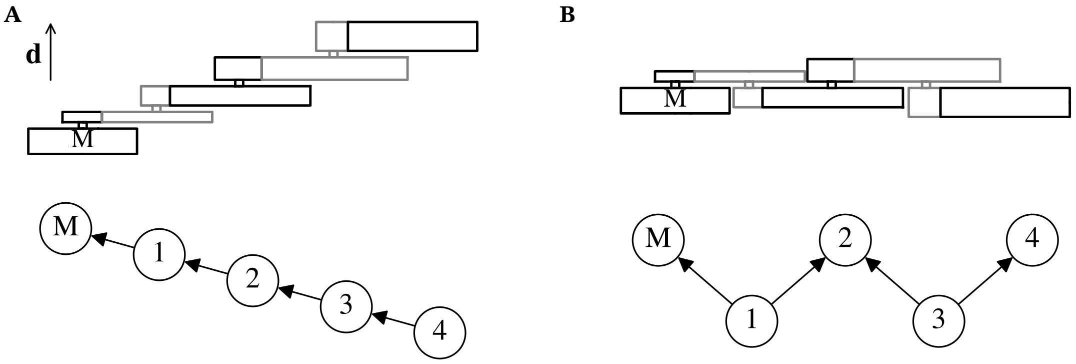


Figure 2.13: Two four-stage actuators examples—(A) a step-like actuator layout and (B) the interlaced actuator layout—and their respective assembly dependency graph, where the shaft groups are numbered successively. Each arrow indicates a dependency.

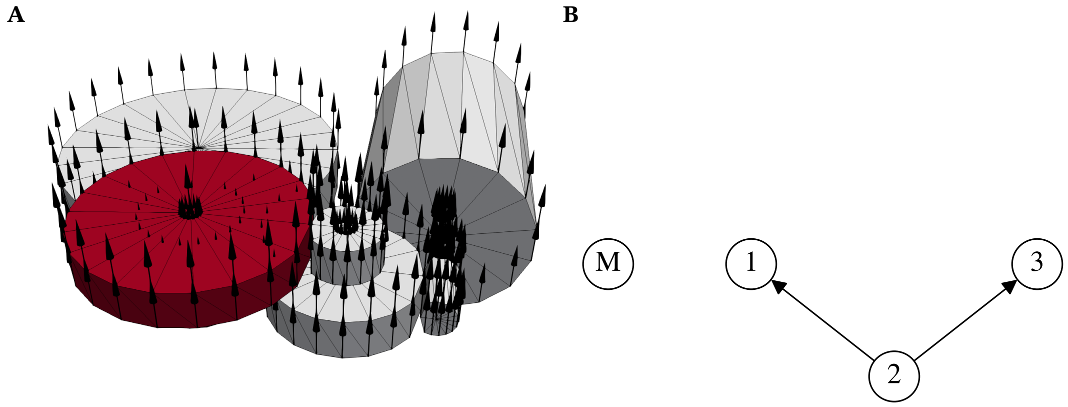


Figure 2.14: (A) Example of an actuator with a crossed helical gear stage and two spur gear stages showing the rays of the assembly evaluation procedure. The group in red corresponds to ②, which can only be inserted after the two gear shafts. (B) The resulting assembly dependency graph, where each arrow indicates a dependency. The motor is not connected, indicating it has no requirement and is not a dependency for any other component.

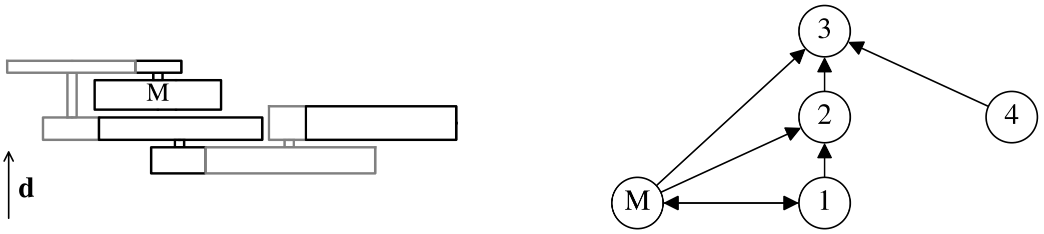


Figure 2.15: Example of a four-stage actuator with an assembly deadlock.

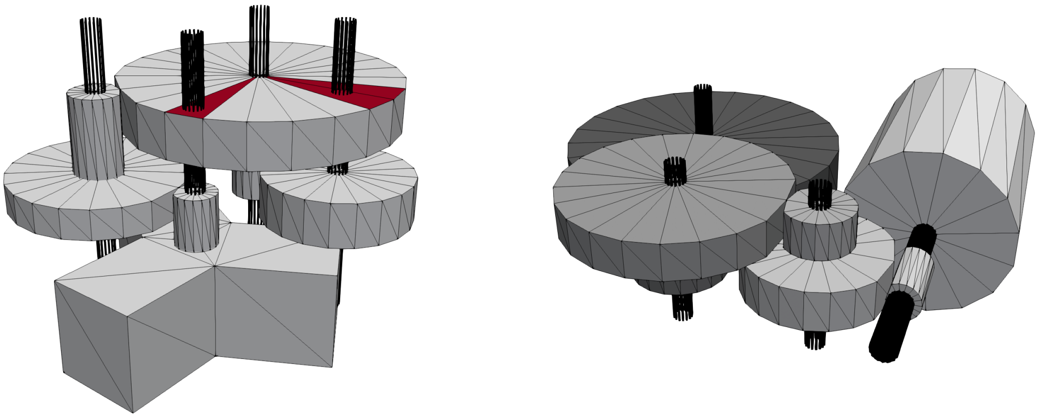


Figure 2.16: Illustrations of the method used to detect if shafts can be supported by the housing for two actuators. Collisions between the rays and gears not belonging to the corresponding shaft group are shown in red.

2.3.4 Shafts supported by the housing

Another way to reduce cost and simplify the actuators is to support the shafts directly with the housing. Indeed, it is easy to create a simple bearing directly in the polymer-injected housing. Further, with proper lubrication, the shaft-housing contact—a polymer metal interface—offers relatively good performance with respect to its complexity. Supporting shafts through the housing implies, however, that the shafts are extended to both sides of the housing, i.e., shafts need to “pass-through”. Such restrictions can have important implications on the design of the gearbox: independently of the required transmission ratio or load, larger gears might be needed to increase their center distance.

Since the idea is to see if shafts interfere with components, ray tracing is well-suited for this task. The method consists in casting a set of rays forming a tube around the location of each shaft and to record collisions with components that are not part of the shaft group. The diameter of the tube can be set on a per-shaft basis, usually 2 to 4 mm. The principle of the method is illustrated in Figure 2.16.

The procedure, detailed in Algorithm 2.4, first collects the coordinates of the center of all shafts and projects them along the direction of the shafts onto the bounding box of the system. A circle with the desired diameter is formed and the rays are cast in the opposite direction. The faces colliding with the rays belonging to another shaft group are counted. The constraint value is obtained by normalizing this face count by the number of rays and the number of groups. In some cases, multiple parts rotating around the same shaft are acceptable, and a “shaft overlap” feature can be enabled. This feature removes the collided faces of groups, whose center coordinates are within one shaft radius of each other. An example of the effect of the “shaft overlap” feature is shown in Figure 2.17.

Algorithm 2.4 Procedure to evaluate if the shafts can be supported by the housing

Require:

$A = \{M_i\}_{i=1}^{n}$ a set of n group meshes
$R = \{r_1, \dots, r_n\}$ the radius for the shaft of each group
$f_{\text{shaft overlap}} \in \{0, 1\}$ the flag for the "shaft overlap feature"

```
procedure PassThroughShafts($A$, $R$, $f_{\text{shaft overlap}}$)
    $B_{\text{box}} \leftarrow$ BoundingBox($A$)    ▷ Get the vertices of the bounding box
    $C \leftarrow \bigcup_{i=1}^{n}$ Center($M_i$)
    $D \leftarrow \bigcup_{i=1}^{n}$ Direction($M_i$)
    $V \leftarrow \bigcup_{i=1}^{n}$ ShaftVertices($\boldsymbol{C}_i, \boldsymbol{D}_i, r_i, B,$ )
                ▷ Project $\boldsymbol{C}_i$ along $\boldsymbol{D}_i$ onto $B$ and generate a circle of radius $r_i$
    $F, R \leftarrow$ RayIntersectAll($A, V, D$)
                ▷ All faces of $A$ intersected by rays cast from $V$ in directions $D$
    $\delta \leftarrow 0$
    $r_{\text{min}} \leftarrow \min R$
    for all $f, r \in F \times R$ do
        $t \leftarrow$ MeshIndex(f)
        $o \leftarrow$ MeshIndex(r)    ▷ Get mesh associated with $f$ or $r$
        if $t \neq o$ then
            $\lambda \leftarrow 1$
            if $f_{\text{shaft overlap}}$ and $\langle \boldsymbol{D}_o, \boldsymbol{D}_t \rangle \in \{-1, 1\}$
               and $\neg$IsMotor($o$) and $\neg$IsMotor($t$) then
                $\boldsymbol{d}_C \leftarrow \boldsymbol{C}_t - \boldsymbol{C}_o$
                $\boldsymbol{d}_\perp \leftarrow \boldsymbol{d}_C - \frac{\langle \boldsymbol{d}_C, \boldsymbol{D}_o \rangle}{||\boldsymbol{d}_C||} \boldsymbol{D}_o$
                $\lambda \leftarrow \max\left\{0, \min\left\{\frac{||\boldsymbol{d}_\perp||_2 - r_{\text{min}}}{r_o + r_t - r_{\text{min}}}, 1\right\}\right\}$
            $\delta \leftarrow \delta + \lambda$
    return $\frac{\delta}{n \cdot |V|}$
```

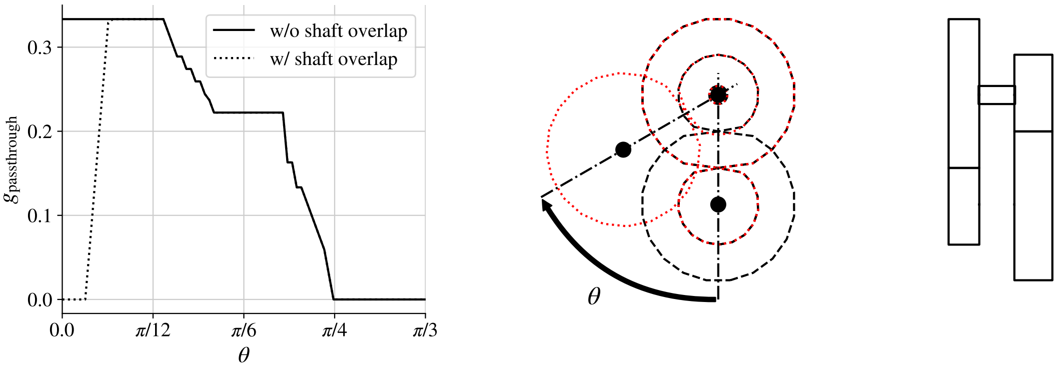


Figure 2.17: Evolution of the pass-through constraint value with and without the "shaft overlap" feature for a two-gear-pair system—shown in top and side views—where the top gear pair is rotated from an initial position, where all axes are aligned, to an angle $\theta = \pi/3$.

2.4 Automated Design Tool

2.4.1 Overview

The presented actuator design framework is versatile and powerful. It is well-suited to serve as the backbone to build an automated design tool capable of designing various actuator configurations considering numerous constraints and performance metrics. By linking the numerical model with (i) a multiobjective optimizer to search for alternatives and to find trade-offs, and (ii) a reporting interface to support decision-making, one is able to build an automated design tool.

The built tool works following the chart shown in Figure 2.18. Engineers and stakeholders analyze and define the specifications that are supplied to the automated design tool. Its process starts by converting the specifications into a numerical optimization problem, that is then solved by a multiobjective optimizer. Reports are regularly stored for engineers to monitor the progress. With these ingredients, the obtained tool is capable of getting well-converged solutions within 30 minutes to 1 hour—on a regular laptop with an Intel Core i7 4 cores at 2.7 GHz. This enables engineers to perform fast iterations including the stakeholders, as envisioned in Figure 1.3.

The conversion step, the multiobjective optimizer, and the reporting interface are briefly described in the following subsections.

2.4.2 Defining the problem

Before being solved by an optimizer, the design problem needs to be formulated as an optimization problem. Typically, optimizer software work with vectors and expect objectives and constraints to follow a certain convention, like the one of Eq. (1.3). Moving from the

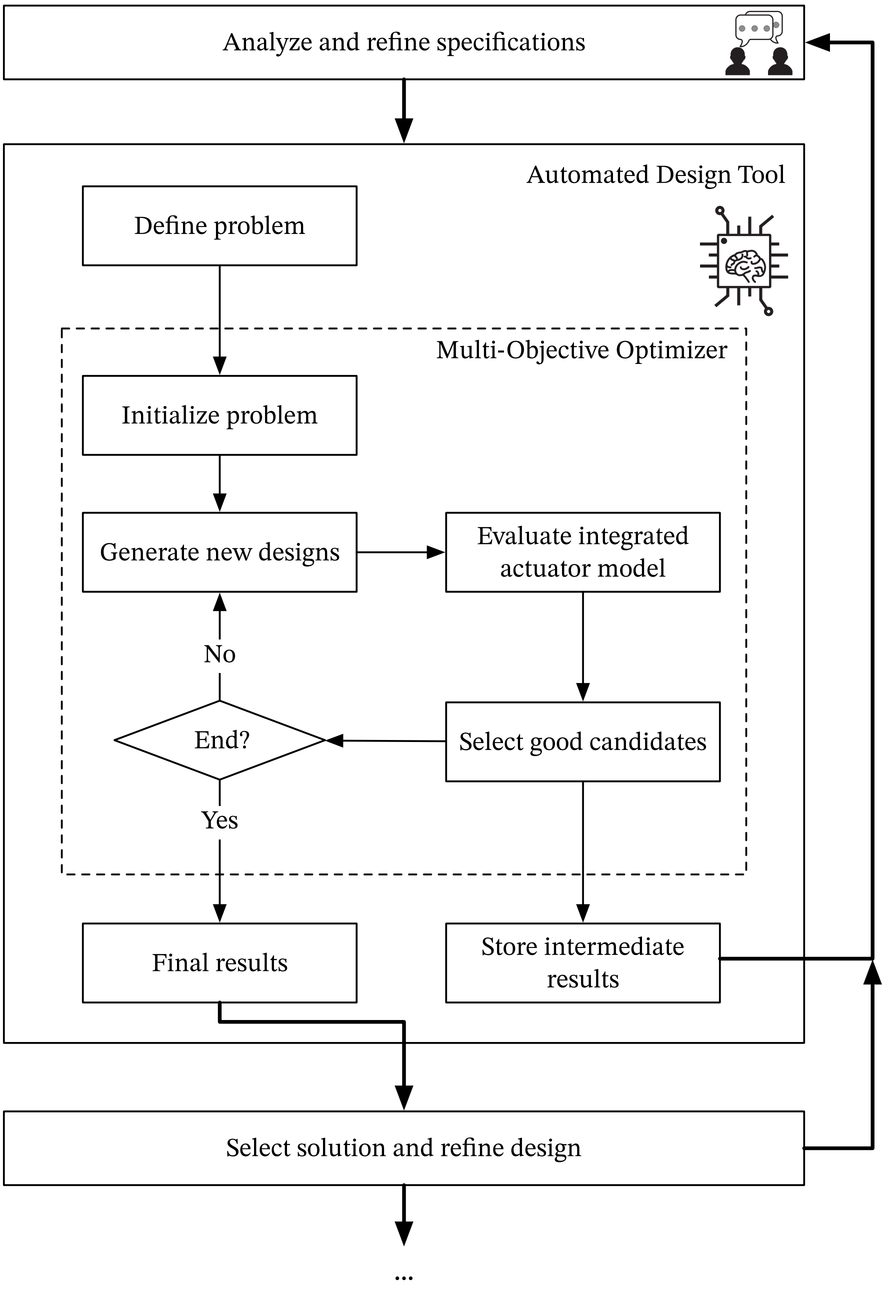

Analyze and refine specifications
Automated Design Tool
Define problem
Multi-Objective Optimizer
Initialize problem
Generate new designs
Evaluate integrated actuator model
No
End?
Select good candidates
Yes
Final results
Store intermediate results
Select solution and refine design
...

Figure 2.18: Overview of the working principle of the automated design tool for actuators.

data representation of the model, which is close to the engineers' way of thinking about actuators, to the representation for an optimizer is a tedious task. This step is therefore taken over by the automated design tool.

Defining a problem requires setting the objectives and constraints, specifying the search bounds of the design variables and listing the desired operating points at which the actuators are simulated. For the objectives, users need to specify which one is to be minimized or maximized, allowing an automatic conversion of all objectives into minimization form. Similarly, constraints are usually defined in the form $g_j(\boldsymbol{x}) \leq b_j$, and are reformulated using Eq. (2.27), as suggested in [87].

$$g_j(\boldsymbol{x}) \leq b_j \Leftrightarrow \begin{cases} \dfrac{g_j(\boldsymbol{x})}{b_j} - 1 \leq 0 & \text{if } b_j > 0 \\ \dfrac{-g_j(\boldsymbol{x})}{b_j} + 1 \leq 0 & \text{if } b_j < 0 \end{cases} \tag{2.27}$$

The most important part, however, is the transformation from the engineers' design variables to the numerical vectors, and vice versa. Engineers specify for each component type the variables to be optimized, and their acceptable range. The motors have up to three design variables: an integer that defines which motor to pick from a predefined list, and two numbers: FF and R_{scale}. The spur gears have up to 14 design variables: two integers selecting their material from a predefined list, two integers $Z_{\{1,2\}}$, and 10 numbers: m, $b^* = b/m$, $x_{\{1,2\}}$, $h^*_{fp,\{1,2\}}$, $\rho^*_{fp,\{1,2\}}$ and $h^*_{ap,\{1,2\}}$. The variables of the crossed helical and planetary gears are given in [165]. Not all variables need, however, to be part of the optimization. Users can set fixed values for certain parameters or let default values be used. For example, the tooth profile A from the ISO norm 53:1998 [80]—$h^*_{fp} = 1.25, \rho^*_{fp} = 0.38, h^*_{ap} = 1.0$—is used unless otherwise specified.

Further, users define a desired actuator configuration. This sets the components to use and their order, e.g., one motor, one crossed helical gear stage, and two spur gear stages. Combining this with the given bounds allows the search space to be calculated. Indeed, the search space corresponds to the union of the design spaces of all components, including the two layout variables d_i and γ_i for each.

Initial experiments suggested that due to the high number of integer variables, the problems were very challenging to solve. To reduce their number, some of them are combined with physically related continuous variables. The resulting number is composed of the integer and its fractional part is mapped to the continuous variable. This is applied for the following:

1. the motor selection integer and the fill factor FF form the combined mFF;
2. the numbers of teeth $Z_{\{1,2\}}$ and their profile shift $x_{\{1,2\}}$ form the combined $Zx_{\{1,2\}}$.

Only the material selection variable remains formulated as an integer variable, but their

Table 2.1: List of crossover and mutation operators used in the adaptive variation operator.

	Crossover operators	
1	Simulated binary crossover [39]	$\eta_c = 20, p_c = 0.9$
2	Simulated binary crossover [39]	$\eta_c = 5, p_c = 0.9$
3	Two-point crossover	
4	Blend crossover [48]	$\alpha = 0.5$
	Mutation operators	
1	Polynomial mutation [39]	$\eta_m = 20, p_m = 1/n$
2	Polynomial mutation [39]	$\eta_m = 5, p_m = 1/n$
3	Uniform mutation	$p_m = 1/n$

number is also reduced by considering that gears of the same shaft share the same material—i.e., they are injected as one part. This assumption corresponds to the reality for the considered applications, but can be deactivated if necessary.

The definition of the design problem by the users is done through:

1. structured text files that are processed by a standard script;
2. custom python scripts.

The first method is the easiest, but offers limited options, since only predefined quantities are calculated and can be included. For full control, custom python scripts need to be written.

2.4.3 Multiobjective optimizer

Despite all code optimization efforts, the evaluation of an actuator design takes 1 ms to 100 ms depending on the activated features. While it might seem little—*it is fast*—when hundreds of thousands of solutions need to be evaluated it still takes a while. In this context, population-based multiobjective evolutionary algorithms (MOEAs) allowing multiple solutions to be evaluated in parallel offer a significant time advantage. In the automated design tool, a modified version of NSGA-II [37] is the main multiobjective optimizer. The base implementation by DEAP [35, 56] has been modified to include an adaptive variation operator inspired by AMALGAM [186] and Borg [68], see Algorithm 2.5, that aims at choosing the best operators at any given moment of the search process. Typically, the exploration probability is set to $\rho_e = 0.2$ and the generic operators in Table 2.1 are considered.

To handle the constraints, different strategies can be activated, but by default, the novel method cEpsilon, described in Chapter 4, is recommended with parameters $\lambda = 0.5$, $c_p = 4$ and $\gamma = 0.6$. Finally, the population size and the number of generations can be chosen by the user—within the constraints of NSGA-II.

Algorithm 2.5 Adaptive variation operator

Require:

$\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ two mating parents
ρ_e the exploration probability
CX and M the lists of available crossover and mutation operators

```
procedure VARIATE($\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\rho_e$, $C$, $M$)
    if RAND < $\rho_e$ then
        crossover ← PICKRANDOMLY($CX$)
    else
        if RAND < 0.5 then
            crossover ← $\boldsymbol{p}_1$.crossover
        else
            crossover ← $\boldsymbol{p}_2$.crossover
    $\boldsymbol{o}_1, \boldsymbol{o}_2$ ← APPLYCROSSOVER(crossover, $\boldsymbol{p}_1$, $\boldsymbol{p}_2$)
    $\boldsymbol{o}_1$.crossover, $\boldsymbol{o}_2$.crossover ← crossover
    for all $\boldsymbol{o} \in \{\boldsymbol{o}_1, \boldsymbol{o}_2\}$ do
        if RAND < $\rho_e$ then
            mutation ← PICKRANDOMLY($M$)
        else
            mutation ← $\boldsymbol{o}$.mutation
        $\boldsymbol{o}$ ← APPLYMUTATION(mutation, $\boldsymbol{o}$)
        $\boldsymbol{o}$.mutation ← mutation
return $\boldsymbol{o}_1, \boldsymbol{o}_2$
```

2.4.4 Interactive result visualization

For the tool to support engineering teams, it needs to (i) store its data in a format readable by machines and humans, (ii) have a traceable process, and (iii) display actuators using familiar representations. A reporting tool has been developed for this purpose. Web technologies are leveraged to build interactive reports in the form of HTML documents that can be opened on any platform and easily shared with colleagues or with the client. The report displays the candidate actuators in four different panels:

1. a scatter plot of the objective values;
2. a parallel coordinate plot with the design variables;
3. a table with the design variables;
4. a WebGL viewer for a CAD-like preview of the actuators.

A screenshot of such a report is shown in Figure 2.19. The reports are generated thanks to the bokeh library[2] [11], which also enables interactivity. Selecting a design in one view, automatically highlights the same design in the other plots. The mesh of an actuator is shown in the WebGL viewer and can be rotated to view it from any desirable angle. Further, the mesh can be downloaded as an STL file—for example to be 3-D printed—or a screenshot can be taken. Since the interactivity is completely managed in the web browser, no software is required.

These reports along with the data as a CSV file are saved regularly during the optimization process. This allows engineers to closely monitor the optimization process and to look at potential results even before the search is completed. This increases the traceability of the process, assists in debugging ill-posed problems and ensures results are available in a timely manner.

2.5 Case studies

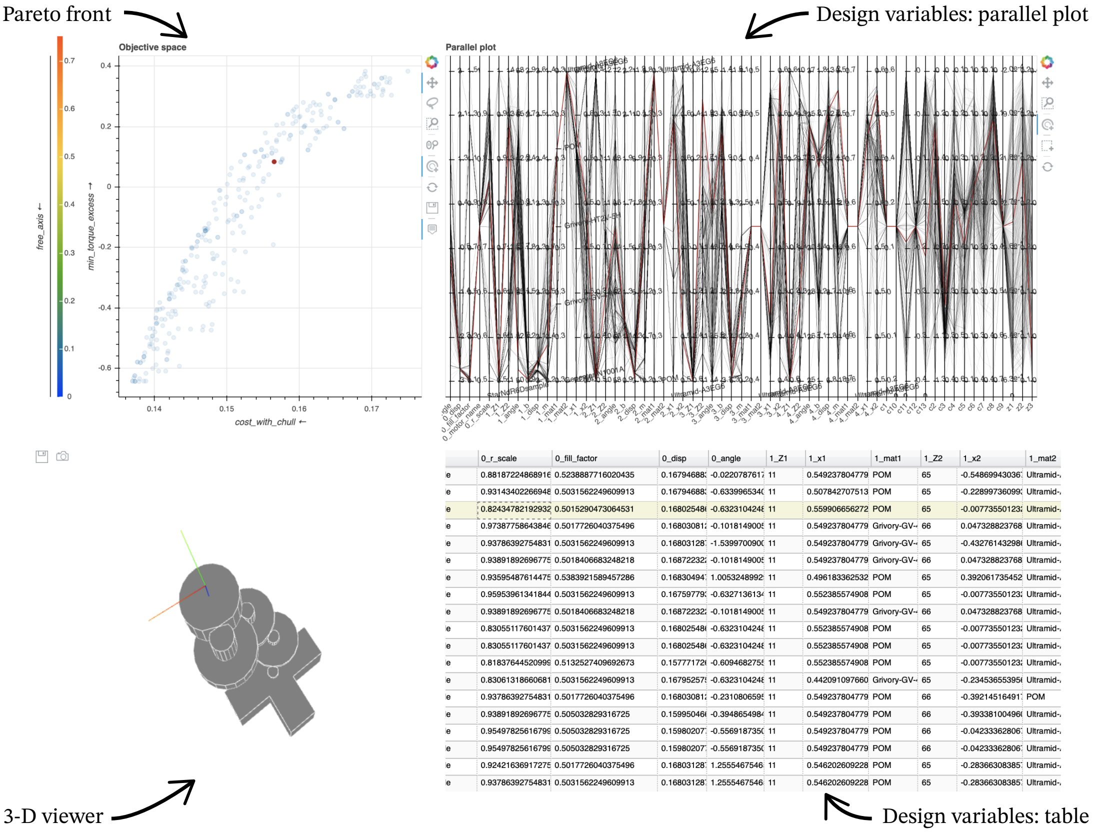
The content of this section has already been published in:

C. Picard and J. Schiffmann. “Automated Design Tool for Automotive Control Actuators”. In: *IDETC-CIE2020*. ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 11B: 46th Design Automation Conference (DAC), Aug. 17, 2020. DOI: 10.1115/DETC2020-22390

Minor adjustments have been made to the text and to the images to match the style of this thesis.

---

[2] https://docs.bokeh.org



	0_r_scale	0_fill_factor	0_disp	0_angle	1_Z1	1_x1	1_mat1	1_Z2	1_x2	1_mat2
e	0.88187224868916	0.5238887716020435	0.16794688	-0.0220787617	11	0.549237804779	POM	65	-0.54869943036	Ultramid-
e	0.93143402266948	0.5031562249609913	0.16794688	-0.633996534	11	0.507842707513	POM	65	-0.22899736099	Ultramid-
e	0.82434782192932	0.5015290473064531	0.16802548	-0.6323104248	11	0.559906656272	POM	65	-0.00773550123	Ultramid-
e	0.97387758643846	0.5017726040375496	0.16803081	-0.1018149005	11	0.549237804779	Grivory-GV-	66	0.047328823768	Ultramid-
e	0.93786392754831	0.5031562249609913	0.16803128	-1.539970090	11	0.549237804779	Grivory-GV-	65	-0.43276143298	Ultramid-
e	0.93891892696775	0.5018406683248218	0.16872232	-0.1018149005	11	0.549237804779	Grivory-GV-	66	0.047328823768	Ultramid-
e	0.93595487614475	0.5383921589457286	0.16830494	1.0053248992	11	0.496183362532	POM	65	0.392061735452	Ultramid-
e	0.95953961341844	0.5031562249609913	0.16759779	-0.6327136134	11	0.552385574908	POM	65	-0.00773550123	Ultramid-
e	0.93891892696775	0.5018406683248218	0.16872232	-0.1018149005	11	0.549237804779	Grivory-GV-	66	0.047328823768	Ultramid-
e	0.83055117601437	0.5031562249609913	0.16802548	-0.6323104248	11	0.552385574908	POM	65	-0.00773550123	Ultramid-
e	0.83055117601437	0.5031562249609913	0.16802548	-0.6323104248	11	0.552385574908	POM	65	-0.00773550123	Ultramid-
e	0.81837644520999	0.5132527409692673	0.15777172	-0.6094682755	11	0.552385574908	POM	65	-0.00773550123	Ultramid-
e	0.83061318660681	0.5031562249609913	0.16795257	-0.6323104248	11	0.442091097660	Grivory-GV-	65	-0.23453655395	Ultramid-
e	0.93786392754831	0.5017726040375496	0.16803081	-0.2310806595	11	0.549237804779	POM	66	-0.39214516491	POM
e	0.93891892696775	0.505032829316725	0.15995046	-0.3948654984	11	0.549237804779	POM	66	-0.39338100496	Ultramid-
e	0.95497825616799	0.505032829316725	0.15980207	-0.556918735	11	0.549237804779	POM	66	-0.04233362806	Ultramid-
e	0.95497825616799	0.505032829316725	0.15980207	-0.556918735	11	0.549237804779	POM	66	-0.04233362806	Ultramid-
e	0.92421636917275	0.5017726040375496	0.16803128	1.2555467546	11	0.546202609228	POM	65	-0.28366308385	Ultramid-
e	0.93786392754831	0.5031562249609913	0.16803128	1.2555467546	11	0.546202609228	POM	65	-0.28366308385	Ultramid-

Figure 2.19: Screenshot of the HTML reporting interface.

Table 2.2: Set of operating points for the HVAC valve application.

ω_{desired}	T_{desired}	Temp.	# cycles	V_s	I_{max}
4.0 rpm	0.4 N m	−40 °C	25 000	13 V	0.2 A
5.5 rpm	0.4 N m	−40 °C	25 000	16 V	0.2 A
4.5 rpm	0.4 N m	23 °C	25 000	13 V	0.2 A
6.0 rpm	0.4 N m	23 °C	25 000	16 V	0.2 A
4.0 rpm	0.4 N m	85 °C	25 000	10 V	0.2 A
2.0 rpm	0.4 N m	85 °C	25 000	13 V	0.2 A

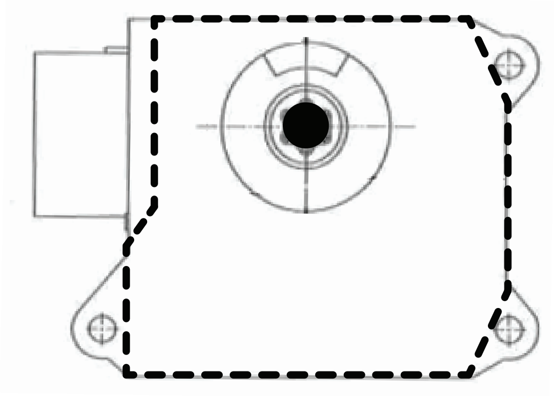


Figure 2.20: Acceptable hull for the HVAC valve actuator application with the position of the output shaft shown as a black dot along with the simplified contour used for the optimizations as a dashed line.

In order to demonstrate the various capabilities of the design tool, two case studies have been selected and explored: an HVAC valve actuator and an HVAC flap actuator. Both actuator types are mounted on the outside of HVAC units of cars and need to be operational at temperatures between −40 °C and 85 °C and for battery voltages between 9 V and 16 V.

In the scenarios considered in this section, the optimization budget has been set to a maximum of 320 000 solution evaluations in order to have good convergence despite the various constraints.

2.5.1 HVAC Valve Actuators

HVAC valve actuators are used in car HVAC systems to control valves that manage the flow of refrigerant in the air conditioning unit. The design problem is to find an actuator that delivers 0.4 N m at a speed of 5 rpm, see Table 2.2 for a complete list of conditions, and that fits inside a given package, see Figure 2.20. In addition, the system should be driven by a DC motor and be irreversible.

The automated design tool is thus configured to search for actuators with a DC motor, a

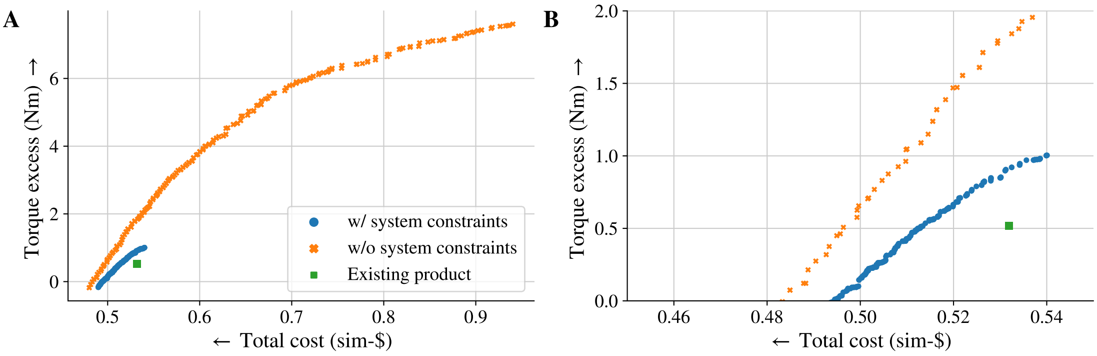


Figure 2.21: Comparison of the proposed designs in terms of total cost and torque excess between an optimization without and with system-level constraints: (A) entire range and (B) zoomed view. The calculated properties of an existing actuator are shown as well.

worm-gear stage and two spur-gear stages. The trade-off between low cost and high output torque is explored, since often performance could be compromised if there is a significant difference in cost.

Based on this case, two scenarios are investigated to illustrate the importance of the system-level constraints: an optimization including all described constraints (including desired packaging) and an optimization without the system-level constraints. The obtained set of solutions for both scenarios along with the characteristics of an existing product—also evaluated by the integrated model—are compared in Figure 2.21.

Since no system dimensions have been included in the first scenario, the automated design tool proposes designs that can generate very large torque, although the gears would never fit in the space constraints. Figure 2.22 shows a design taken from this optimization with comparable performance to the existing actuator. Obviously, substantial work is still needed to package the candidate design appropriately with no guarantee that it is actually possible.

Including system-level constraints allows the tool to directly produce useful design candidates and to determine the maximum torque, an actuator contained within the imposed packaging, can deliver. The identification of the trade-off between output torque and cost helps engineers in their discussion with a client. Compared to the existing product, the tool suggests an actuator 3.6% cheaper with a similar torque or 69% more torque for the same cost. For both designs, the shafts can all be supported directly by the housing and they both respect the assembly and packaging constraints.

Figure 2.23(A) shows the solution with a similar torque output as the existing product. Not only is it cheaper, it is also sensibly smaller. These are key arguments suggesting that the algorithm is capable of finding competitive solutions even when compared to designs that have been improved over several years.

In terms of design, this candidate actuator has a gearbox with a ratio of 165, which is sig-

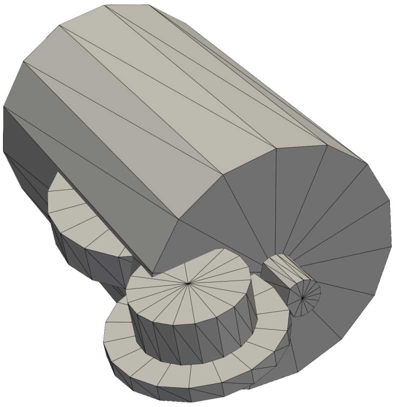


Figure 2.22: Example of a design returned by the tool when not considering layout constraints.

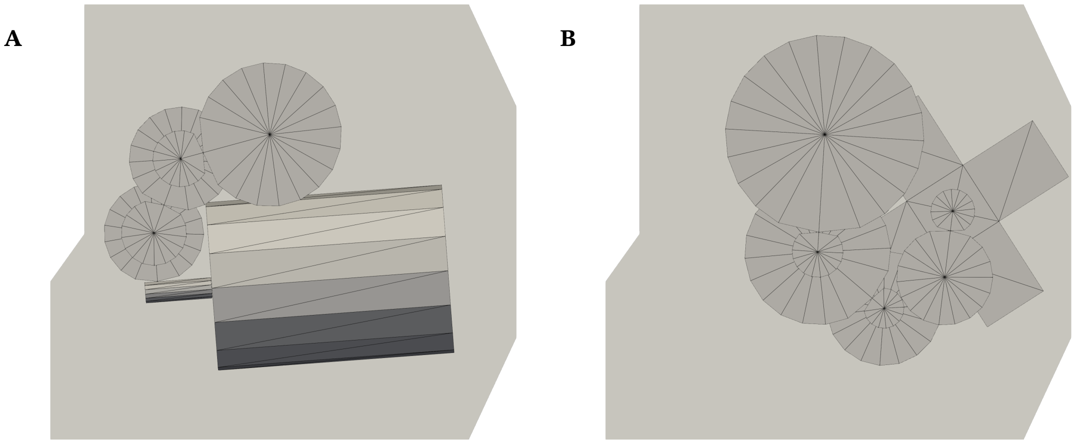


Figure 2.23: Comparison of two alternate designs that match the requirements for an HVAC valve actuator using (A) a DC motor or (B) a stepper motor.

nificantly smaller than the ratio of 419 of the original product. This is an important shift in the equilibrium between the motor and the gearbox, that is easily assessed in an integrated design setting. Here, the motor runs at lower speeds, which may also help reduce noise and wear of the brushes in the motor.

Finally, since the tool is so fast, designs with different technological choices can be generated rapidly and used to suggest viable alternatives to the client. For example, Figure 2.23(B) shows that it is also possible to fit an actuator driven by a stepper motor with four stages of spur gears in the same package and achieve equal performance. The tool thus allows to explore the full design space and gives tangible support to decision-makers.

Table 2.3: Set of operating points to evaluate HVAC flap actuators.

Output	T_{desired}	Temp.	# cycles	V_s	$I_{\max}$	Condition
$0.314\,\mathrm{rad\,s^{-1}}$	0.3 N m	−40 °C	10 000	9 V	0.15 A	running
$0.314\,\mathrm{rad\,s^{-1}}$	0.3 N m	25 °C	50 000	9 V	0.15 A	running
$0.314\,\mathrm{rad\,s^{-1}}$	0.3 N m	40 °C	20 000	9 V	0.15 A	running
$0.314\,\mathrm{rad\,s^{-1}}$	0.3 N m	80 °C	10 000	9 V	0.15 A	running
0.0056 rad	1.21 N m	−40 °C	1000	9 V	0.32 A	holding
0.0056 rad	1.21 N m	25 °C	5000	9 V	0.32 A	holding
0.0056 rad	1.21 N m	40 °C	2000	9 V	0.32 A	holding
0.0056 rad	1.21 N m	80 °C	1000	9 V	0.32 A	holding

2.5.2 HVAC flaps actuators

The second case study focuses on the design of an HVAC flap actuator. They are used to control the orientation of flaps to adjust the mixture between hot and cold air before it enters the interior of cars. The design problem is to find an actuator that delivers 0.3 N m at 3 rpm and 1.21 N m in holding condition, see Table 2.3 for a complete list of conditions. Stepper motors should be used and there is a required minimal resolution of 6400 steps per turn at the output. The package information for this application is shown in Figure 2.10(right). For these optimizations, all system-level constraints are considered: packaging, assembly capabilities and shaft supported by the housing. Further, the diameter used to validate the clearance of the output shaft is set to 6 mm to have enough space to include the linkage plug of the flap directly in the output gear.

Actuators with three, four and five stages of spur gears are evaluated using the automated design tool. The corresponding Pareto fronts are shown in Figure 2.24. Note that the actuators with five stages slightly violate the shaft clearance constraint. Further, the actuators with four stages offer the best trade-off given the limited space. For a performance similar to the existing actuator, cost can be reduced by 11% or torque can be increased by 170% while also cutting cost by 7.7%.

Figure 2.25 compares the layout of the existing product to a candidate actuator with a similar performance and shows that the latter is more compact, eliminating the need for the protrusion at the bottom for the electrical connector. The engineering team can still further work on the proposed design, for example, by rotating the whole motor-gear system around the output shaft to better balance the margins with the different sides of the housing.

In terms of design, the proposed actuator has a transmission ratio of 485 and a height of 11.5 mm compared to 561 and 14.5 mm for the existing product. The gains in cost come mostly from smaller coils for the motor.

In order to better understand the impact of the housing-supported-shaft constraint, another

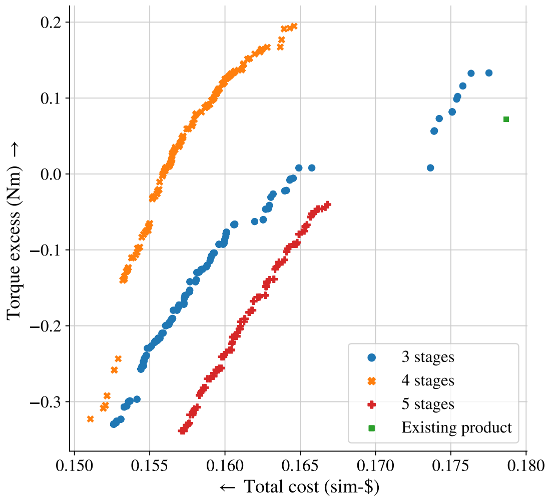


Figure 2.24: Comparison between the obtained Pareto fronts for HVAC flap actuators with three, four, and five stages of spur gears.

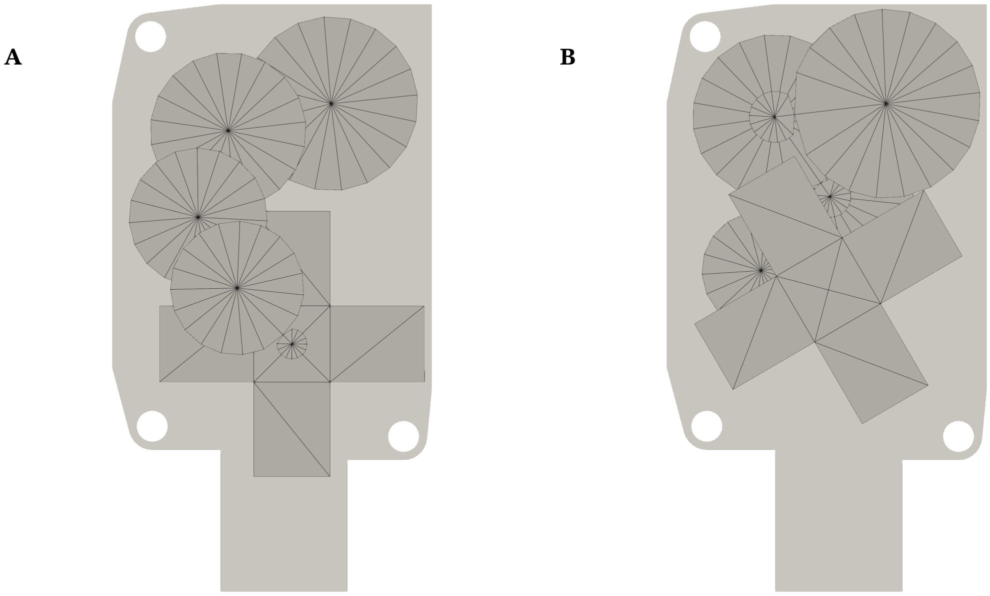


Figure 2.25: Comparison of the layout of (A) the existing flap actuator and (B) a candidate actuator generated by the automated design tool.

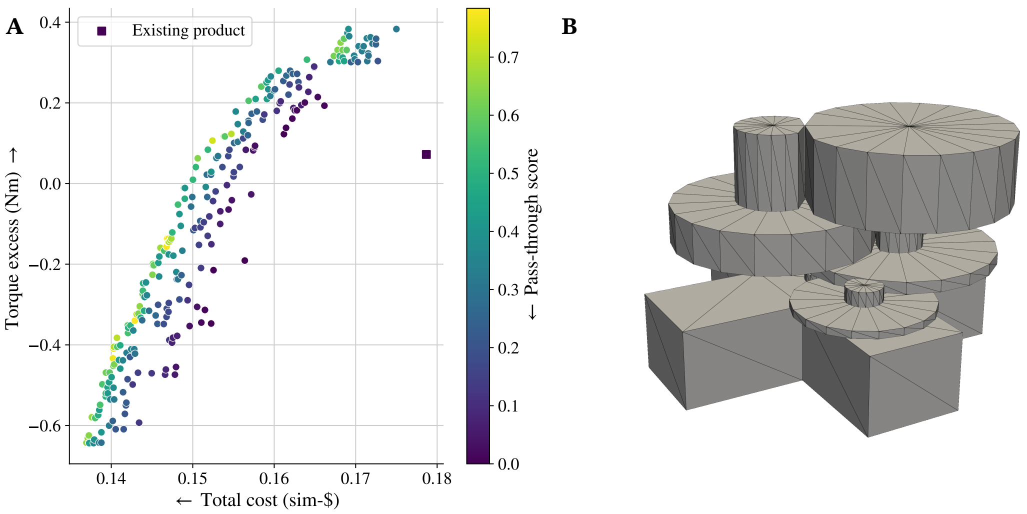


Figure 2.26: (A) Pareto front of the flap actuator design problem when the shaft pass-through constraint is set as an objective and (B) an example of a compact actuator design.

optimization in which this constraint is now considered as a third objective, has been performed. The obtained Pareto front is shown in Figure 2.26(A) and emphasizes the important role this constraint plays in the previous optimizations. By relaxing this constraint, more compact and even cheaper (-6%) actuators can be considered. Figure 2.26(B) shows what such an actuator would look like. It would now be the work of the engineering team to find solutions to effectively support the shafts in such a compact design.

Through the use of 3-D meshes in the automated design tool, candidate designs exist in a tangible form that can be understood and apprehended by anyone. In order to further promote discussion between engineers and customers, actuators can be 3-D printed making it easy to show around and comment, see Figure 2.27.

2.6 Optimization challenges

As good results as shown in the case studies are possible thanks to the work described in Chapters 3 and 4, and in particular to the novel cEpsilon constraint handling strategy (CHS). The use of the original CHS for NSGA-II in the automated design tool presented one important challenge for its practical use: optimization outcome reliability. To highlight this challenge, Figure 2.28 shows the fronts obtained by running NSGA-II with its original CHS four times on the same optimization problem. The differences between the outcomes are important. Further, the more constrained the design problem, the more the original method struggles to find any feasible solutions.

It serves as motivation to investigate why a state-of-the-art algorithm performs so poorly,

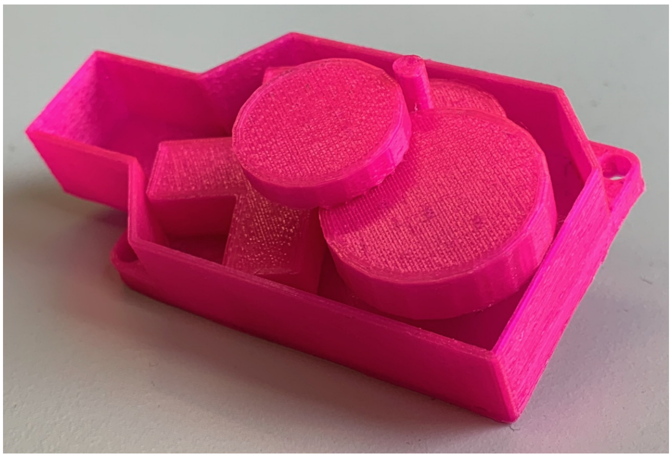


Figure 2.27: Picture of a 3-D printed candidate design.

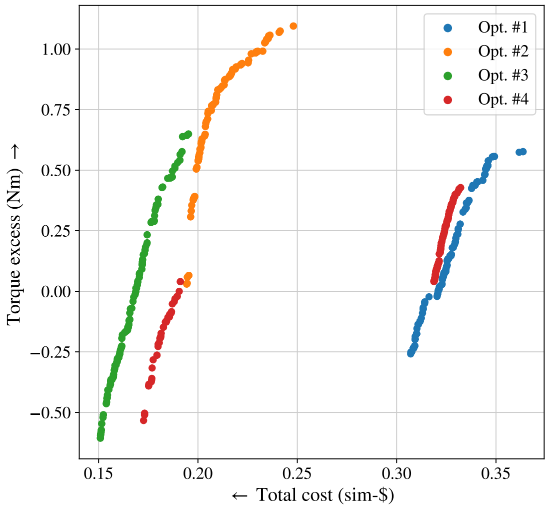


Figure 2.28: Fronts obtained for the HVAC flap actuator design problem for four different optimization runs.

and how MOEAs can be improved to better work on such design problems. The design problems that are being solved in this chapter are certainly complex, but they are quite simple if one thinks about extending this approach to systems like heat pumps, airplanes or artificial hearts.

2.7 Concluding remarks

In this chapter, a framework for the design of geared electro-mechanical actuators has been introduced. It relies on fast component models that are assembled like bricks to form a whole system. Methods and procedures to simulate important system-level specifications are formulated. They tackle diverse challenges related to packaging that are typical for these systems. With these constraints, the obtained numerical model is an integrated model since it simulates multiple components and disciplines in a single pass.

This versatile model serves as the core of an automated design tool, whose aim is to speed up the design of actuators, to provide the engineering team and the stakeholders with a better picture of the trade-offs, and to support well-informed decisions. The automated design tool achieves this by linking the integrated model with multiobjective optimization and interactive reports.

The potential is highlighted through two real case studies from the automotive industry: the design of actuators for HVAC valves and HVAC flaps. In both cases, the tool proposed interesting and valuable alternatives, outperforming the current products in terms of cost and available power, while respecting the packaging specifications. These results were obtained within an hour on a regular laptop.

The case studies highlight the importance of the presented system-level constraints. Indeed, the candidate designs obtained, if those constraints are not considered, are of significantly less interest to the engineers, exaggerating the achievable performance, while leaving the engineers with the complex and repetitive work of laying out the components. And while optimization challenges related to the CHS of NSGA-II have been identified, they are addressed in the coming chapters, opening up the way to a tool to thoroughly investigate many design alternatives.

The potential of the framework is much greater than the two case studies shown in the work. There are possible extensions of the integrated model, for example to include the use of standard components from a database or to evaluate complete series of actuators sharing components. Both additions can be game-changing for actuator manufacturer, enabling great economies of scale. Further, the simultaneous optimization of configuration and parameters of actuators offers many challenges and opportunities, including a broader reflection about optimality in this context. The Master thesis of Lemaitre [102], performed under the author's supervision, discusses some aspects of this vast topic.

Questa o quella per me pari sono
A quant'altre d'intorno mi vedo

Il Duca di Mantova (Rigoletto, G. Verdi)

3

Realistic Benchmark Problems

Designing 'good' multiobjective optimization algorithms and selecting the appropriate ones for a given class of problems require that common evaluation procedures are defined. For this purpose, researcher typically follow recommended methodologies [94] and use so-called benchmark problems, making their results comparable among each other. These benchmark optimization problems should typically be representative of various challenges found in actual problems.

Most benchmark constrained multiobjective optimization problems (CMOPs) currently used are synthetic problems [87, 202], i.e., mathematical formula created for the purpose of testing multiobjective optimizers. While their advantages are that they are fast to evaluate and that their properties and Pareto fronts can be explicitly calculated, they tend to have rather unrealistic features. Indeed, recently reported results [142, 173, 174, 200] including the results presented in Section 2.6 suggest that although state-of-the-art optimization algorithms have good performance on benchmark problems, they fail at reliably solving simple 'real-world' problems. As such, more realistic benchmark problems are needed to promote the development of multiobjective optimization (MOO) that can be successfully applied in engineering.

Further, better problem characterization is needed to have a proper understanding of the reasons that lead certain algorithms to perform differently, but determining the properties of many engineering problems is not straight-forward. Their objectives and constraints are resulting from expensive simulations and cannot be explicitly solved. Therefore, appropriate numerical methods are needed.

In this chapter, existing benchmark problems and numerical methods to characterize problems are introduced. Then, a realistic benchmark framework, called Multi-Objective Design of Actuators (MODAct), is presented. The core is a simplification of the numerical model from Chapter 2, from which 20 CMOPs with up to five objectives and adjustable constraints are derived. Their characteristics are evaluated with an extended constraint landscape analysis and compared to existing CMOPs from literature.

The content of this chapter has already been published in:

C. Picard and J. Schiffmann. “Realistic Constrained Multiobjective Optimization Benchmark Problems From Design”. In: *IEEE Transactions on Evolutionary Computation* 25.2 (Apr. 2021), pp. 234–246. DOI: 10.1109/TEVC.2020.3020046

Sections II and IV to VI have been included as is into Sections 3.1 and 3.3 to 3.5.
©IEEE 2021. Reprinted, with permission, from the authors.

Figures have been regenerated to match the style and layout of this thesis.

Author contributions: CP and JS designed research; CP performed research, and wrote the article himself.

3.1 Related work

3.1.1 Constrained multiobjective optimization benchmark problems

Various CMOPs have been proposed over the years. Some of the oldest are the SRN [166], the TNK [175] and the OSY [133] functions. They have been commonly used to benchmark algorithms [37, 194]. These are two objective problems with few decision variables—two for SRN and TNK, and six for OSY—and the same number of inequality constraints. Mostly due to the low dimensionality of their decision space, their complexity is low [41].

Based on this observation, Deb et al. proposed a new scalable framework for CMOPs [41]. Functions derived from this framework have, in theory, an unbounded dimensionality and their complexity can be adjusted by the selection of a helper function. This framework has been used to derive the CTP test suite [41] with 7 functions (CTP1 to CTP7), which are typically limited to two objectives, two decision variables and one or two inequality constraints and remains therefore quite simple. The framework is also the base of the CF test suite [202] created for the IEEE CEC2009 MOEA Competition. Compared to CTP, CF problems have a larger search space ($n = 10$) and three out of the ten functions have three objectives. The results of the competition showed that some of these functions are difficult to solve. The number of constraints is also low (two inequality constraints). More recent variants of this framework have been proposed (e.g., NCTP [104]), but the shortcomings mentioned previously remain.

With the need to solve optimization problems with an increasing number of objectives, many-objective optimizers have been developed along with appropriate test problems. The DTLZ test suite [42], which is scalable also in terms of objectives, has been extended with constraints to form the C-DTLZ test suite [87]. There are three inequality constraint sets (C1, C2 and C3) that can be combined with the unconstrained DTLZ functions (e.g., C1-DTLZ1 or C1-DTLZ3). C1 and C2 types add one constraint, while the C3 type adds one constraint per objective. While promising, Tanabe and Oyama showed that even algorithms discarding the constraints could solve some of the C-DTLZ problems [173].

In order to represent real-world problems better, Ma and Wang proposed a new framework to build test functions with more inequality constraints and a large infeasible search space [112]. They derived the MW test suite with 14 instances with up to four inequality constraints and a feasability ratio (FsR) very close to zero. Three instances are scalable in terms of objectives and they cover various front geometries.

The DAS-CMOP and DAS-CMaOP test suites [51] introduce the concept of tunable constraints through a difficulty triplet (η, ζ, γ), with $\eta, \zeta, \gamma \in [0, 1]$. DAS-CMOP is composed of nine base problems with two or three objectives and 11 or 7 constraints. DAS-CMaOP adds another nine base problems for $m > 3$, extending the WFG framework [78] with $2m + 1$ similarly tunable constraints. 16 given difficulty triplets are suggested, four of which result in equality constraints. In total, this generates 288 test functions, 144 of which are scalable in terms of objectives. This massive test suite offers a great potential that has yet to be evaluated.

Given the strong ties between optimization and engineering applications, real-world like problems have also been proposed as benchmark problems [38, 40, 87, 142, 148]. Among those, the car-side impact [87] and the water problem [38, 148] have more than two objectives. Their decision space is relatively small, but they have a large number of constraints (10 and 7 respectively). It has been shown, however, that most solutions generated during optimization only violate one of the constraints and that unconstrained NSGA-II could also solve them fairly well [173].

Based on these considerations, the following benchmark CMOPs have been selected for this study: the CTP and CF functions to allow for comparison with prior work, the recent MW test suite and a subset of DAS-CMOP with the hard difficulty triplets 9, 10, 11 and 12 and the car-side impact and water problems as existing real-world like problems. They are summarized in Table 3.1.

3.1.2 Constraint landscape analysis

The complexity of CMOPs is often discussed using the feasibility ratio, the number of constraints and some descriptive adjectives (e.g., multi-modal, non-linear, rugged, active,...) [107, 173, 194]. The limitations of these means to characterize complexity has already been suggested by various researchers [115, 173]. With C-DTLZ, Jain and Deb [87] proposed a classification scheme for constraints based on the changes they introduce with respect to the unconstrained problem:

- Type-1 constraints introduce an “infeasible barrier” in the objective space, but the Pareto front is not affected.

Table 3.1: Number of objectives, search variables and constraints of the selected benchmark CMOPs along with their FsR—calculated as explained in Section 3.3 ©IEEE 2021

	m	n	p	q	FsR
CTP1	2	2	2	0	0.997
CTP2	2	2	1	0	0.990
CTP3	2	2	1	0	0.989
CTP4	2	2	1	0	0.967
CTP5	2	2	1	0	0.989
CTP6	2	2	1	0	0.493
CTP7	2	2	1	0	0.643
CF1	2	10	2	0	0.521
CF2	2	10	2	0	0.994
CF3	2	10	2	0	1.000
CF4	2	10	2	0	0.503
CF5	2	10	2	0	0.512
CF6	2	10	2	0	0.307
CF7	2	10	2	0	0.319
CF8	3	10	2	0	0.004
CF9	3	10	2	0	0.160
CF10	3	10	2	0	0.000
MW1	2	15	1	0	0.000
MW2	2	15	1	0	0.000
MW3	2	15	2	0	0.000
MW4	≥ 3	$12 + m$	1	0	0.000
MW5	2	15	3	0	0.000
MW6	2	15	1	0	0.000
MW7	2	15	2	0	0.000
MW8	≥ 3	$12 + m$	1	0	0.000
MW9	2	15	1	0	0.000
MW10	2	15	3	0	0.000
MW11	2	15	4	0	0.000
MW12	2	15	2	0	0.000
MW13	2	15	2	0	0.002
MW14	≥ 3	$12 + m$	1	0	0.000
DAS-CMOP3_9	2	30	11	0	0.334
DAS-CMOP3_10	2	30	11	0	0.000
DAS-CMOP3_11	2	30	11	0	1.000
DAS-CMOP3_12	2	30	11	0	0.000
DAS-CMOP6_9	2	30	11	0	0.333
DAS-CMOP6_10	2	30	11	0	0.000
DAS-CMOP6_11	2	30	11	0	1.000
DAS-CMOP6_12	2	30	11	0	0.000
DAS-CMOP8_9	3	30	7	0	0.111
DAS-CMOP8_10	3	30	7	0	0.000
DAS-CMOP8_11	3	30	7	0	1.000
DAS-CMOP8_12	3	30	7	0	0.000
Car-side impact	3	7	10	0	0.181
Water	5	3	7	0	0.920

- Type-2 constraints make a part of the unconstrained Pareto front infeasible.
- Type-3 constraints make the full region of the unconstrained Pareto front infeasible and the location of the Pareto front is governed by the constraints.

While, this classification and other similar ones [51, 112] offer insights into the effects of the constraints in the objective space, they do neither quantify these effects nor do they consider the changes due to the constraints in the search space.

In recent work on constrained single-objective continuous and combinatorial optimizations, Malan et al. [114, 115] propose to look at the constraints as defining a “violation landscape” that can be analyzed much in the same fashion as the fitness landscape. This landscape can be described in the search space using the FsR and the ratio of feasible boundary crossing ($RFB_\times$) and in the objective space with the fitness violation correlation (FVC) and the ideal zone (IZ) metrics. The definition of these metrics is given as follows:

- The $RFB_\times$ measures the proportion of steps, which imply crossing the feasibility boundary on a progressive random walk [113] through the search space and quantifies the disjoint nature of the feasible space.
- The FVC is the Spearman’s rank correlation between the fitness and the constraint violation CV (3.1), measuring the contradiction between the objective and the constraints.
- The IZ quantifies the proportion of points present in the good unconstrained fitness, low violation zone of the fitness-violation plot and represents the likelihood of finding points in that zone.

$$CV(\boldsymbol{x}) = \sum_{j=1}^{p} \langle g_j(\boldsymbol{x}) \rangle + \sum_{k=1}^{q} |h_k(\boldsymbol{x})| \tag{3.1}$$

$$\text{with } \langle \alpha \rangle = \begin{cases} \alpha & \text{if } \alpha > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Malan et al. [115] applied this approach on the constrained single-objective optimization problems of the CEC2010 competition [117] and were able to link the achieved performance of the competing algorithms to the score of these metrics. Thus, they show the potential of this method to better characterize problems and partially address the algorithm selection problem.

While the FsR and the $RFB_\times$ can be translated directly to MOO, the FVC and IZ require a single numerical fitness value, which is ambiguous in a multiobjective context. Two alternate metrics adapted to multiobjective problems will be presented in Section 3.3.

3.2 Multi-Objective Design of Actuators (MODAct)

The framework for the design of electro-mechanical actuators presented in Chapter 2 is used in this context to derive a test suite. Yet, in order to make the code easier to use by other researchers, a reduced version of the framework is created. While it may seem counter-intuitive to use “simplified” models when wanting to define challenging CMOPs, the results from the following sections will show that they nevertheless offer sufficient complexity for their purpose. The reduced framework is called MODAct and its source code is available online[1]. Like the original framework, MODAct is written in Python, but additional interfaces to common programming languages (C++ or MATLAB) are proposed and allow the use of many optimizers. In particular, examples showing how to use it with the following optimizer software are provided: Borg [68], NOMAD [101], PlatEMO [176] and pymoo [9].

Compared to the original model, MODAct has a few limitations:

1. Only two-phase steppers and steel spur gears are available;
2. The torque prediction uses only Eq. (2.1) with static and dynamic friction and the motor parameters are taken from five existing steppers;
3. Fatigue and temperature dependent effects are removed from the gear model;
4. Only cylindrical meshes are used both for the motors and the gears;
5. The layout constraints relying on ray-tracing are omitted.

With these changes, the computational time to evaluate one actuator (including objectives and constraints) is of about 20 ms on 2016 laptop. This offers the advantage that it is both fast enough to follow traditional benchmarking methodologies, while allowing researchers interested in the development of parallelization and distributed computation approaches to also test their algorithms.

3.2.1 Definition of the various problems

Using MODAct and inspired by the industrial applications from Section 2.5, 20 optimization problems can be built. The aim of these optimization problems is to find suitable three-stage actuators ($k = 3$) that operate at two operating points, see Table 3.2. For simplicity, the tooth profile A is used and the material of the gears is set to steel.

Five possible objectives are considered:

- minimize total cost (C), Eq. (2.26);
- maximize minimum torque excess for each considered operating point (T), Eq. (2.23);

---

[1] https://github.com/epfl-lamd/modact

Table 3.2: Operating point requirements for all MODAct problems ©IEEE 2021

#	Rotational speed	Desired torque	Voltage	Max current
1	1.35 rad/s	0.6 N · m	9 V	2 A
2	0.3 rad/s	1 N · m	12 V	2 A

- maximize harmonic mean of the safety factors to bending S_F and to pitting S_H for all gears (S), Eq. (2.22);
- maximize electrical to mechanical energy conversion efficiency (E), Eq. (2.24);
- minimize transmission ratio (I), Eq. (2.19).

From these objectives, five possible combinations are considered and identified by grouping the capital letters: CS, CT, CTS, CTSE, CTSEI. Each forms a problem class.

Further, 11 common constraints are considered to be combined with these problem classes. They fit into five categories:

- gear constraints ($\epsilon_\gamma \geq 1.1$, $\zeta_{f1,2} \geq -5$, no interference, $S_F, S_H \geq 1$);
- required minimum torque excess;
- bounding box dimensions limited to $bb_y \leq 50$ mm and $bb_z \leq 35$ mm;
- output shaft within 5 mm of a desired location.

Four levels of constraint complexity (1, 2, 3 and 4) are created using these constraints. In practice, which level is selected depends upon the development stage or the application. In this context, it allows to offer a variety of challenges for optimizers, since each level can be associated with all previously defined problem classes. In total, 20 unique benchmark problems are available, see Table 3.3. Each problem is named by appending the constraint complexity level to the objective class (class CTSE combined with constraint level 3 forms problems CTSE3). Important to note, constraints are mostly unaffected by the chosen objectives, expect for the required minimum torque excess (two different settings).

Table 3.3: Summary of the 20 functions of MODAct including details about their constraints and the search space ©IEEE 2021

	Constraint level	n	m	p	q	Gear constraints (6)	Min. torque excess (1)	3-D Collisions (1)	Bounding box (2)	Output shaft (1)	Search space bounds $[mFF, R_{scale}$ $Zx_{11}, Zx_{12}, m_1, b_1, d_1, \gamma_1,$ $Zx_{21}, Zx_{22}, m_2, b_2, d_2, \gamma_2,$ $Zx_{31}, Zx_{32}, m_3, b_3, d_3, \gamma_3]$
CS1	1	20	2	7	0	Yes	≥ -0.001	No	No	No	$\boldsymbol{x}^{(L)} = [0, 0.3,$ $9, 30, 0.3, 5, -20, -\pi,$ $9, 30, 0.3, 5, -20, -\pi,$ $9, 30, 0.3, 5, -20, -\pi]$
CS2	2	20	2	8	0	Yes	≥ -0.001	Yes	No	No	
CS3	3	20	2	10	0	Yes	≥ -0.001	Yes	Yes	No	
CS4	4	20	2	9	0	Yes	≥ -0.001	Yes	No	Yes	
CT1, CTS1, CTSE1, CTSEI1	1	20	2, 3, 4, 5	7	0	Yes	≥ -0.599	No	No	No	
CT2, CTS2, CTSE2, CTSEI2	2	20	2, 3, 4, 5	8	0	Yes	≥ -0.599	Yes	No	No	$\boldsymbol{x}^{(U)} = [5 - 10^{-6}, 2,$ $41 - 10^{-6}, 81 - 10^{-6}, 1, 15, 20, \pi,$ $41 - 10^{-6}, 81 - 10^{-6}, 1, 15, 20, \pi,$ $41 - 10^{-6}, 81 - 10^{-6}, 1, 15, 20, \pi]$
CT3, CTS3, CTSE3, CTSEI3	3	20	2, 3, 4, 5	10	0	Yes	≥ -0.599	Yes	Yes	No	
CT4, CTS4, CTSE4, CTSEI4	4	20	2, 3, 4, 5	9	0	Yes	≥ -0.599	Yes	No	Yes	

3.3 Multiobjective constraint landscape analysis

Calculating the objectives and constraints of such an actuator involves several steps. It is therefore not straight-forward to predict their mathematical model characteristics. Thus, MODAct problems need to be analyzed with metrics characterizing the effect of the constraints on both the search and objective spaces.

This section presents the metrics for the constraint landscape analysis introduced by [115], as well as additional methods suited for multiobjective problems. The analysis relies on both repeated independent uniform samplings and progressive random walks [113] of the decision space.

Definition 3.1. Given an independent uniform sampling $\mathcal{U} \subset \mathcal{S}$, the FsR is defined as:

$$\mathrm{FsR} := \frac{|\{\boldsymbol{u} \in \mathcal{U} \mid \mathrm{feasible}(\boldsymbol{u})\}|}{|\mathcal{U}|} \tag{3.3}$$

where feasible($\boldsymbol{x}$) is an indicator function indicating if a solution $\boldsymbol{x}$ is feasible, i.e., $CV(\boldsymbol{x}) = 0$.

Definition 3.2. Given a sequence of s samples $\mathcal{W}$ generated by a random walk of $s-1$ steps $\mathcal{W} = \{\boldsymbol{w_1}, \boldsymbol{w_2}, \dots, \boldsymbol{w_s}\}$, the $\mathrm{RFB}_\times$ is defined as:

$$\mathrm{RFB}_\times := \frac{1}{s-1} \sum_{i=1}^{s-1} \chi(i) \tag{3.4}$$

$$\chi(i) = \begin{cases} 0 & \text{if } \mathrm{feasible}(\boldsymbol{w_i}) = \mathrm{feasible}(\boldsymbol{w_{i+1}}) \\ 1 & \text{otherwise} \end{cases} \tag{3.5}$$

where the helper function χ indicates when the feasibility boundary is crossed.

The $\mathrm{RFB}_\times$ metric should be high for a disjoint feasible space and low for a contiguous feasible space. Yet, the possible values of $\mathrm{RFB}_\times$ also depend on the number of feasible points s_f encountered during a walk $\mathcal{W}$, thus making the comparison between functions difficult. In order to compare, one would need to know, given the ratio of encountered feasible points per walk, how disjoint the space is. In other words, one wants to identify the maximum possible $\mathrm{RFB}_\times$ and define this as an upper bound for a given ratio s_f/s. In general, this is equivalent to trying to spread out the feasible or infeasible points (depending on which ones are the minority) in a sequence to maximize the number of transitions. Following this approach, it can be shown that the upper bound of $\mathrm{RFB}_\times$ for a given walk is:

$$\mathrm{RFB}_{\times,\max} = \frac{2}{s-1} \cdot \min\{s_f, s - s_f, \frac{s-1}{2}\} \tag{3.6}$$

This upper bound is used to define the normalized ratio of feasible boundary crossing ($\mathrm{nRFB}_{\times}$):

$$\mathrm{nRFB}_{\times} = \begin{cases} 0 & \text{if } \mathrm{RFB}_{\times} = 0 \\ \frac{\mathrm{RFB}_{\times}}{\mathrm{RFB}_{\times,\max}} & \text{otherwise} \end{cases} \tag{3.7}$$

An $\mathrm{nRFB}_{\times}$ value of 1 is obtained for walks in the search space that are as disjoint as possible given their s_f/s ratio.

For the analysis of the objective space, two new metrics are introduced: PFd and PFcv. The main idea is to capture the interactions between the image in the objective space of randomly selected points in the search space and the Pareto fronts. Both metrics rely on the previously obtained aggregated set of samples $\mathcal{L} = \mathcal{U}_1 \cup ... \cup \mathcal{W}_1 \cup ...$ and the Pareto front $\mathcal{PF}^*$ of the problem.

PFd is constraint independent and represents the average minimal distance from the Pareto front to the cloud of points formed by $\boldsymbol{f}(\mathcal{L})$. It measures the ease of randomly generating points near the Pareto front. In order to make it comparable between functions, the points in the objective space are normalized using the ideal $\boldsymbol{z}^*$ and nadir $\boldsymbol{z}^{\text{nad}}$ vectors of $\mathcal{PF}^*$, leading to the normalized objective function $\boldsymbol{f}^n$ and the normalized Pareto front $\mathcal{PF}_n^*$. PFd corresponds to the inverted generational distance (IGD) [25] between $\mathcal{PF}_n^*$ and $\boldsymbol{f}^n(\mathcal{L})$.

Definition 3.3. Given a set of samples $\mathcal{L}$ and their normalized image $\boldsymbol{f}^n(\mathcal{L})$ and given the normalized Pareto front $\mathcal{PF}_n^*$, PFd is defined as:

$$\mathrm{PFd} := \frac{1}{|\mathcal{PF}_n^*|} \sum_{\boldsymbol{z} \in \mathcal{PF}_n^*} \min_{\boldsymbol{l} \in f^n(\mathcal{L})} ||\boldsymbol{z} - \boldsymbol{l}||_2 \tag{3.8}$$

where $||\cdot||_2$ is the Euclidean norm.

PFcv measures the average constraint violation CV value of neighbors of the Pareto front in the sample set. This represents the sensitivity in terms of constraints of solutions on the Pareto front.

Definition 3.4. Given a set of samples $\mathcal{L}$, the Pareto front $\mathcal{PF}^*$ and K the number of neighbors to consider, PFcv is defined as:

$$\mathrm{PFcv} := \frac{1}{CV_{95}|\mathcal{PF}^*|} \sum_{\boldsymbol{z} \in \mathcal{PF}^*} \frac{1}{K} \sum_{\boldsymbol{x} \in \mathcal{B}(\boldsymbol{z},K)} CV(\boldsymbol{x}) \tag{3.9}$$

where CV_{95} is the 95th percentile of the CV values found in $\mathcal{L}$ and $\mathcal{B}(\boldsymbol{z}, K)$ is the set of the K closest neighbors of $\boldsymbol{z}$ in $\mathcal{L}$.

3.4 Methods of the numerical investigations

MODAct problems are compared to five groups of benchmark problems identified from literature in Section 3.1: the CTP, the CF, the MW and some DAS-CMOP functions and two real-world like problems (water and car-side impact). The comparison is done based on the presented constraint landscape analysis approach and on a convergence study. The next sections present the parameters used for the various steps.

3.4.1 Constraint landscape analysis

Both uniform samplings and progressive random walks have been performed using the parameters mentioned in [115]: 30 independent uniform sampling $\mathcal{U}$ of $|\mathcal{U}| = 1000n$ points and $30n$ independent progressive random walks $\mathcal{W}$ of 1000 steps ($|\mathcal{W}| = s = 1001$) each with maximum step size of 1% of the decision space. The reported FsR, $\text{RFB}_\times$ and $\text{nRFB}_\times$ scores are obtained by averaging over all independent samplings.

The PFd and PFcv metrics are calculated with the best-known Pareto front obtained from the convergence study when the true Pareto front is unknown. The $K = 20$ closest neighbors are considered for PFcv.

3.4.2 Convergence study

In addition to these characteristics, the main interest is to investigate how well these design optimization problems can be solved and understand how the different constraint levels influence convergence. This is achieved by performing a convergence study where the problems are compared among each other and against two kinds of optimizers: the commonly used algorithms NSGA-II [37] and NSGA-III [36] on one side and C-TAEA [105] a recent algorithm developed specifically to tackle CMOPs on the other side. NSGA-II and NSGA-III follow a feasibility first approach and rely in the survival step on the constrained-dominance principle (CDP) introduced in [37]:

Definition 3.5. Given two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, $\boldsymbol{x}_1$ is said to constrained-dominate $\boldsymbol{x}_2$, if one of the following is true:

1. $\boldsymbol{x}_1$ is feasible and $\boldsymbol{x}_2$ is not;
2. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are infeasible and $CV(\boldsymbol{x}_1) < CV(\boldsymbol{x}_2)$;
3. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are feasible and $\boldsymbol{x}_1$ (Pareto-)dominates $\boldsymbol{x}_2$.

C-TAEA maintains two separate archives—one for diversity, the other for convergence—and has a special restricted mating approach to balance between the two.

The study is performed using pymoo [9]. NSGA-II is used for all problems with $m = 2$ and

Table 3.4: Parameters used to configure each run of NSGA-II, NSGA-III and C-TAEA ©IEEE 2021

Parameter	NSGA-II/III	C-TAEA
Population size μ	200	210 or 220 ($m = 4$)
Number of function (i.e., solution) evaluations	300 000	
Mutation η_m	20	
Mutation rate	$1/n$	
Crossover η_c	15	
Crossover probability (CXPB)	0.9	

NSGA-III for all problems with more objectives. Each optimization is performed 30 times with the parameters specified in Table 3.4, following common practice for these algorithms. The number of reference directions for NSGA-III is chosen as close to the population size, while following Das and Dennis's approach [33]. The same approach is used for C-TAEA, except the population size and number of reference directions are set to the same value. An unbounded external archive (UEA) is added to the algorithms, as recommended in [12], to collect all feasible non-dominated solutions along the optimization.

For MODAct problems, the solutions of the archives from all runs are aggregated and sorted to determine the best-known Pareto front. The estimated ideal $\boldsymbol{z}^*$ and nadir $\boldsymbol{z}^{\text{nad}}$ points are collected to provide per-problem front normalization.

The convergence and diversity are evaluated with the hypervolume indicator [204]. In particular, the exact and fast implementations by the Walking-Fish Group are used [28, 189]. Since the objective functions of the various problems have different scales and since there is a mix of minimization and maximization objectives, the Pareto front and the archives are transformed into minimization only problems and normalized with the corresponding $\boldsymbol{z}^*$ and $\boldsymbol{z}^{\text{nad}}$. Only then the hypervolume is calculated using a common reference point $\boldsymbol{r} = (1.1, \dots, 1.1)^T$. Finally, the comparison is made through the relative hypervolume error ΔHV_n (3.10) with respect to the best-known Pareto front:

$$\Delta HV_n = \frac{HV(\mathcal{PF}^*_{m,n}, \boldsymbol{r}) - HV(\mathcal{A}_{m,n}, \boldsymbol{r})}{HV(\mathcal{PF}^*_{m,n}, \boldsymbol{r})} \tag{3.10}$$

where $\mathcal{A}$ denotes a given external archive and the subscript $_m$ that the problems have been converted to minimization.

For the statistical difference between optimization algorithms, the non-parametric Wilcoxon rank-sum test with the null hypothesis that all algorithms are equal is applied with a confidence interval of 99%.

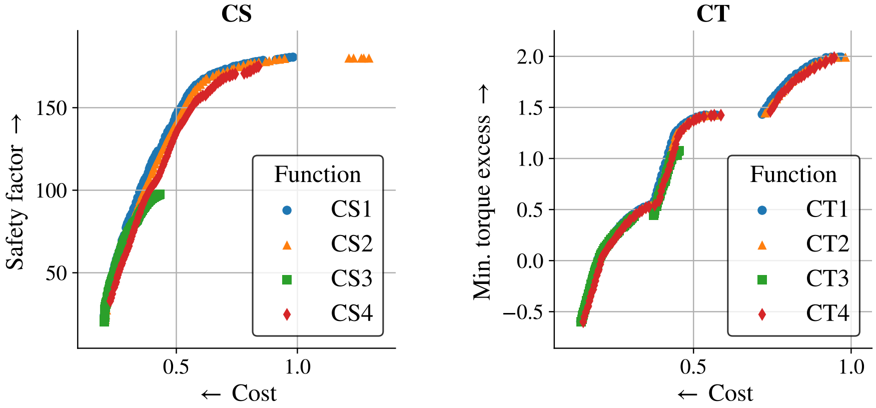


Figure 3.1: Best-known Pareto fronts for (A) CS and (B) CT problems

3.5 Results and discussion

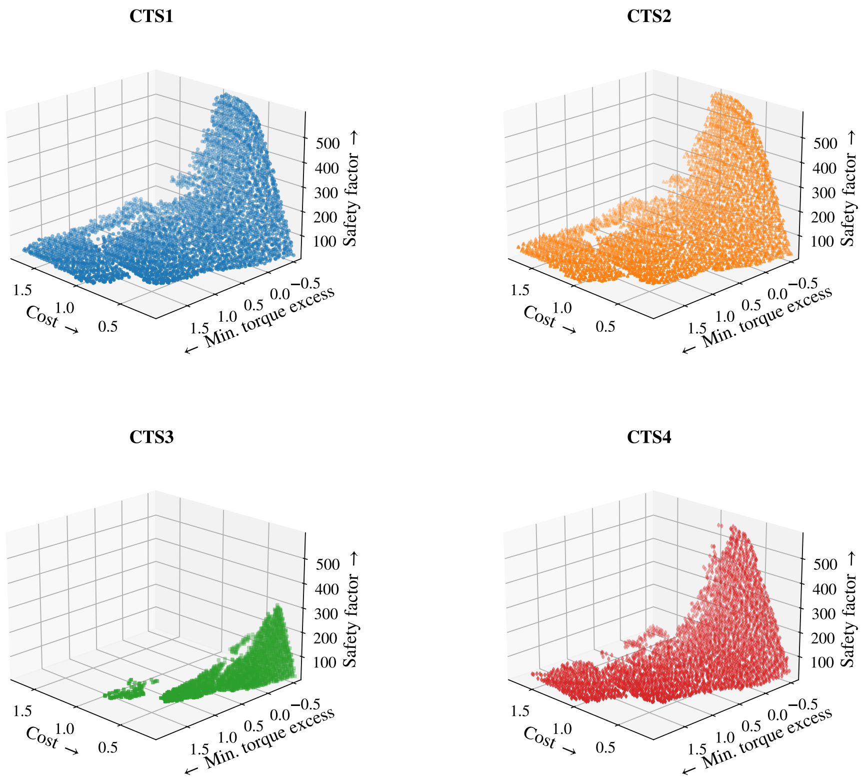
3.5.1 Design trade-offs and constraints

The analysis of MODAct problems starts by looking at some of the best-known Pareto fronts. All best-known Pareto fronts can be downloaded [143]. Figure 3.1 shows the best-known Pareto fronts for CS and CT problems, while Figure 3.2 represents the CTS problems. The best-known Pareto fronts of CS problems consist of a smooth convex segment, while CT problems have several step-like disconnected segments. CTS problems combine the two features to form a complex surface with concave/convex and disconnected parts.

The discontinuities are indeed expected for these design problems since some variables represent discrete physical choices. In particular, the motor selection is dominating these effects. With different cost and power ranges, the motor designs do not overlap, leading to the visible steps. This feature is common in design problems and should therefore also appear in benchmark problems. For engineers, this is of particular interest since these steps imply important design trade-offs.

Looking more closely at the effects of the constraints, it can be noticed, that constraint levels 2 and 4 slightly shift the found fronts towards higher costs compared to level 1. Constraint level 3 is the most restrictive: the limit on the size of the bounding box of the actuator does not allow large gears or motors, which significantly reduces the available options. While the impact is limited in the objective space, this is not the case in the decision space. An analysis of neighbors in the objective space between CT1 and CT2 confirms important differences, mostly on the number of teeth of the wheels Z_{i2} and the spatial positioning variables d_i and γ_i.

For problems with more objectives, the best-known Pareto front of CTSEI3 is shown as an example in Figure 3.3 projected into the various objective planes. It shows that there are indeed up to five competing objectives with different shapes.


CTS1

CTS2

CTS3

CTS4

Figure 3.2: Best-known Pareto fronts for CTS problems

3.5.2 Convergence analysis

In a second step, the ease of convergence and the repeatability are compared. Figure 3.4 represents box plots of the obtained relative hypervolume errors (3.10) over the 30 optimization runs for each problem by NSGA-II/III and by C-TAEA. Starting with MODAct problems, the results suggest the following:

1. The proposed instances pose a wide range of optimization challenges to NSGA-II/III and C-TAEA.
2. Within the same classes, constraint levels 1 and 2 are generally equally well solved with rare outliers that may exhibit early convergence to local optima.
3. Constraint levels 3 and 4 are increasingly difficult and the various optimization runs achieve very different levels of convergence.
4. There is also a clear trend of increasing hypervolume error with the number of objectives, but this is certainly related to the nature of the indicator itself.
5. The more restrictive threshold on the minimum torque excess of CS problems seems to negatively impact convergence for all constraint levels. In particular, C-TAEA is

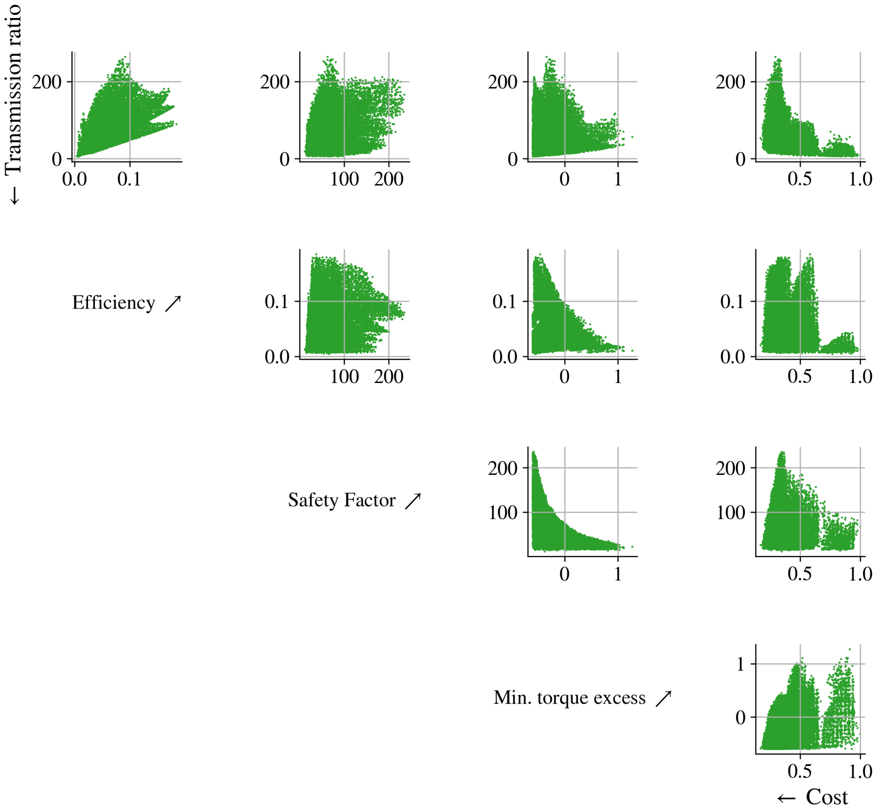


Figure 3.3: Best-known Pareto front for CTSEI3, for rendering questions, the numbers of points displayed has been reduced by eliminating points that were too close to each other

barely able to find any solutions within the boundaries of the reference point for CS3 and CS4.

In comparison, the CTP, car-side impact, water, most MW and DAS-CMOP8 problems are effectively solved by the considered algorithms. MW10, MW11 and MW13 are the most difficult of MW and can be compared to MODAct constraint levels 1 and 2. The CF family offers a broader range of challenging problems. The findings that the biobjective problems CF3, CF5 and CF7 are the hardest are consistent with the outcomes of the CEC2009 MOEA Competition [201]. Among the three-objective problems, CF8 and CF10 are the most challenging in particular for NSGA-III. The DAS-CMOP3 problems seem to be challenging for both algorithms. DAS-CMOP6 problems are effectively solved by C-TAEA, while NSGA-II struggles for DAS-CMOP6_11 and DAS-CMOP6_12.

In terms of algorithms, NSGA-II/III is overall significantly better than C-TAEA (better on 30 problems, no difference on 18 and worse on 17), despite using a simple constraint handling strategy. NSGA-II/III is always better for CTP and for the car-side impact problem, although the difference is minor. The results are more balanced for the other benchmark problems. For the MW test suite, C-TAEA performs better on the difficult problems. C-TAEA also

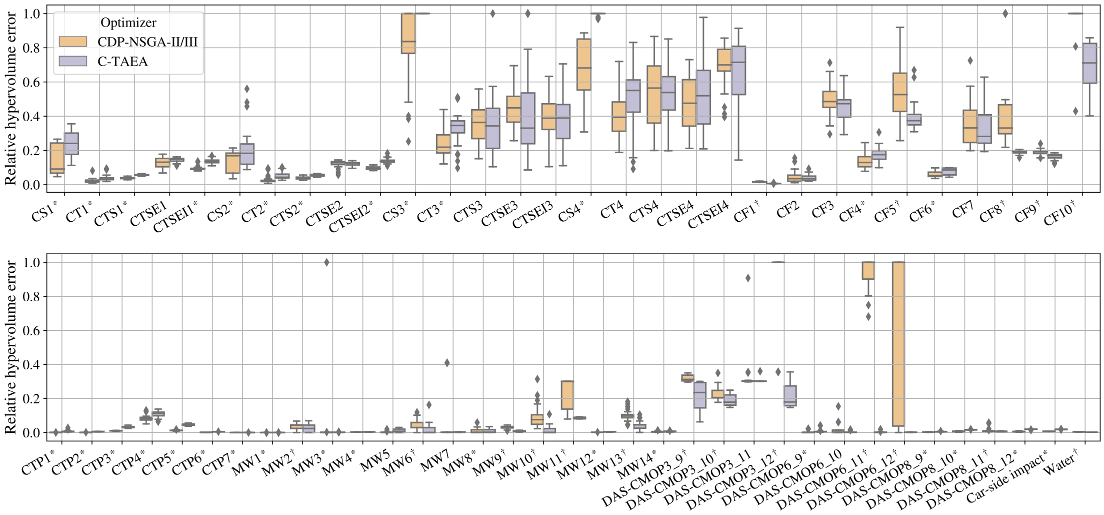


Figure 3.4: Box plot of the relative hypervolume errors ΔHV_n of the external archives obtained for each problem by NSGA-II/NSGA-III versus C-TAEA, where the $*$ or † after the function name is used to indicate that C-TAEA is, respectively, significantly worse or better than NSGA-II/NSGA-III

performs better on the three-objective CF problems and is on a par for biobjective problems. The opposite is true for DAS-CMOP: NSGA-III is better for three-objective problems, while C-TAEA shows solid performance on biobjective instances. Finally on MODAct problems NSGA-II/III has a clear advantage for biobjective problems and is at least as good as C-TAEA otherwise.

In general, the performance of NSGA-II/III and C-TAEA is insufficient on MODAct instances with constraint levels 3 and 4, although they represent very common and simple mechanical design problems. The observed large variance of the optimization outcomes has important practical consequences. Considering problems CS3 and CS4, more than 75% of the optimization runs of NSGA-II obtain approximate Pareto fronts with a hypervolume of 50% or smaller than the best-known Pareto fronts, while C-TAEA fails to find interesting solutions. As an example, the best run of NSGA-II and of C-TAEA along with two optimization runs are compared to the best-known Pareto front for problem CS4 in Figure 3.5 to illustrate the large difference in the proposed solutions with a lack of convergence and diversity. Using partially converged solutions for decision-making can lead to significantly different engineering outcomes.

In order to better understand the optimization process of MODAct problems, the evolution of the relative hypervolume error of the external archives of NSGA-II, C-TAEA and NSGA-II without constraint handling are compared in Figure 3.6. For all four problems, the unconstrained optimization clearly fails to get many feasible solutions, confirming that the constraints play a key role in MODAct problems. While for problems CT1 and CT2 the optimization budget is more than sufficient, better results may be possible for CT3 and CT4.

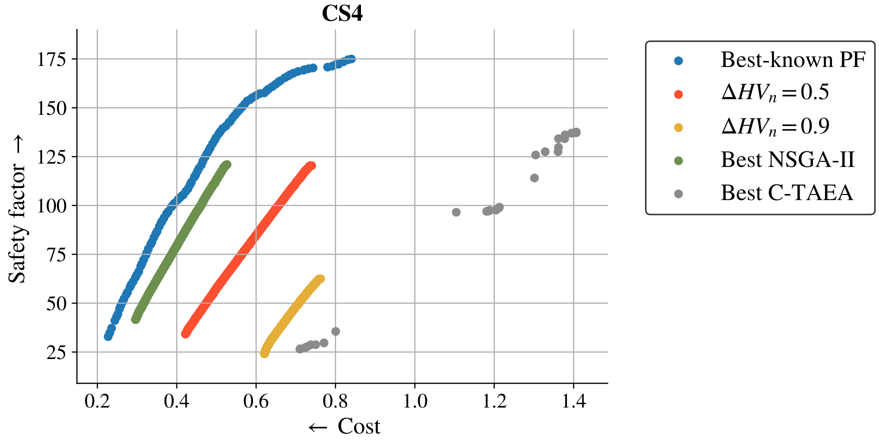


Figure 3.5: Comparison of the approximate Pareto fronts obtained by the best runs for NSGA-II and C-TAEA and two runs achieving a ΔHV_n of 0.9 and 0.5 against the best-known Pareto front for CS4

Yet, while the mean relative hypervolume error seems to be decreasing, the spread remains almost constant (NSGA-II) or increasing (C-TAEA). Running the optimization algorithms longer might thus not necessarily address the repeatability issue. It is also interesting to note that while CT3 and CT4 are challenging problems, feasible solutions are rapidly found: on average after 800 evaluated solutions for CT3, and after 2000 for CT4.

3.5.3 Link between convergence and constraints

Finally, the objective is to use the results from the constraint landscape analysis to identify the underlying characteristics that affect convergence most and how the newly introduced problems differ from the existing benchmarks functions.

The obtained metrics for all problems are summarized in Table 3.5. To begin, the advantages of $\mathrm{nRFB}_\times$ are evaluated. CF1, CTP6 and CTP7 have the three highest $\mathrm{RFB}_\times$ scores. They also have an FsR close to 0.5. For problems with a high or a low FsR, the definition of $\mathrm{RFB}_\times$ necessarily decreases its possible values, thus masking the level of discontinuity of small infeasible or feasible spaces. In such cases, the $\mathrm{nRFB}_\times$ metric acts as an amplifier of the scores through the normalization. The $\mathrm{nRFB}_\times$ values indicate that all CTP problems have a relatively disjoint search space, which is known a priori from their definition [41]. MODAct problems have $\mathrm{nRFB}_\times$ values ranging between the CTP and the CF problems, suggesting a rather disjoint search space. The same applies to DAS-CMOP problems with difficulty triplet 9, DAS-CMOP3_11 and MW13. $\mathrm{nRFB}_\times$ values can also remain low such as for the car-side impact and water problems, suggesting a contiguous feasible space. It is noted that some limitations remain, in particular for highly feasible or infeasible search spaces, where boundaries are hard to find. This is the case for the rest of DAS-CMOP and MW.

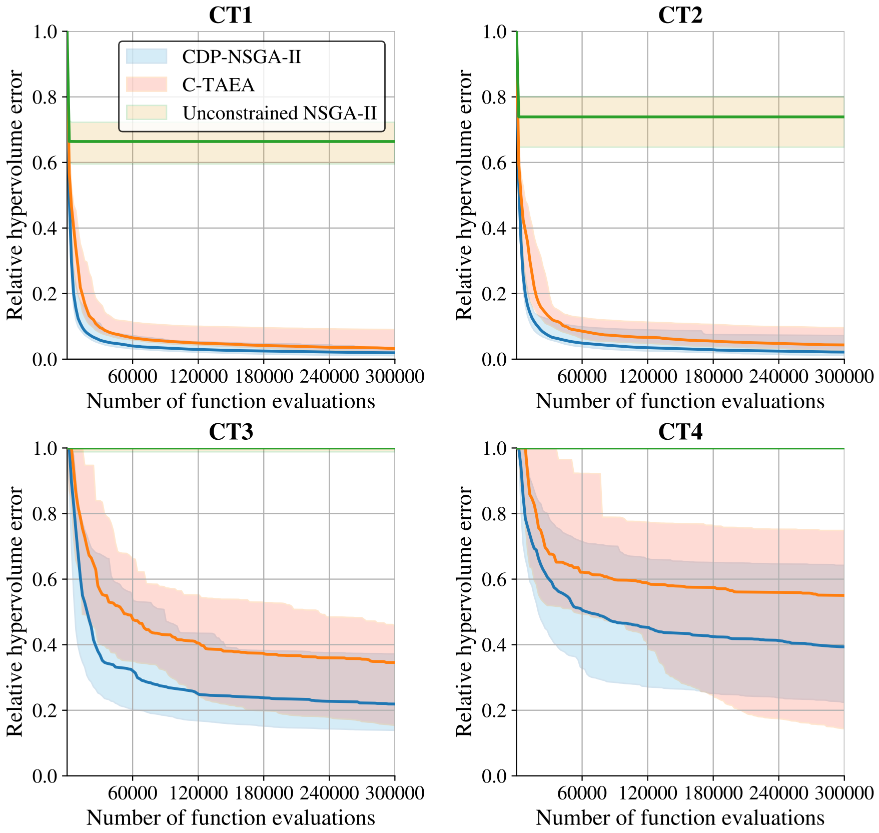


Figure 3.6: Evolution of the median relative hypervolume error including 5th and 95th percentiles for CT problems comparing: NSGA-II with the constrained-dominance strategy (CDP-NSGA-II), C-TAEA and NSGA-II discarding constraints (Unconstrained NSGA-II)

Table 3.5: Calculated metrics resulting from the constraint landscape analysis ©IEEE 2021

Function	FsR	$RFB_\times$	$nRFB_\times$	PFd	PFcv
CS1	0.0022	0.0016	0.1155	0.0015	0.4158
CT1	0.0342	0.0278	0.3081	0.0071	1.7115
CTS1	0.0343	0.0281	0.3095	0.0788	0.8317
CTSE1	0.0342	0.0283	0.3050	0.1235	0.7521
CTSEI1	0.0347	0.0287	0.3075	0.1369	1.2519
CS2	0.0010	0.0007	0.0593	0.0021	0.4703
CT2	0.0160	0.0156	0.3097	0.0068	1.6274
CTS2	0.0158	0.0158	0.2947	0.0796	0.7688
CTSE2	0.0160	0.0161	0.3075	0.1242	0.7226
CTSEI2	0.0162	0.0161	0.3110	0.1411	1.1255
CS3	0.0000	0.0000	0.0000	0.0012	0.6763
CT3	0.0000	0.0001	0.0126	0.0029	2.2518
CTS3	0.0000	0.0001	0.0112	0.0551	0.9973
CTSE3	0.0000	0.0001	0.0178	0.0643	1.5531
CTSEI3	0.0000	0.0001	0.0259	0.0336	8.4287
CS4	0.0000	0.0000	0.0000	0.0013	0.7802
CT4	0.0000	0.0000	0.0000	0.0039	0.7694
CTS4	0.0000	0.0000	0.0000	0.0483	0.7807
CTSE4	0.0000	0.0000	0.0000	0.0569	0.9021
CTSEI4	0.0000	0.0000	0.0000	0.0277	1.6211
CF1	0.5199	0.1539	0.2007	0.0301	0.7801
CF2	0.9942	0.0033	0.1682	0.1114	0.0116
CF3	1.0000	0.0000	0.0000	1.9014	0.0000
CF4	0.5001	0.0154	0.0169	0.3835	0.1432
CF5	0.5068	0.0080	0.0090	2.8356	0.0574
CF6	0.3056	0.0089	0.0222	0.4670	0.0306
CF7	0.3151	0.0169	0.0325	2.5682	0.0896
CF8	0.0042	0.0012	0.0427	0.7449	0.1984
CF9	0.1597	0.0340	0.1259	0.3445	0.2047
CF10	0.0001	0.0001	0.0298	2.8237	0.1391
CTP1	0.9972	0.0048	0.7769	0.0429	0.0001
CTP2	0.9899	0.0100	0.4902	0.0505	0.0771
CTP3	0.9892	0.0100	0.4659	0.0481	0.0653
CTP4	0.9656	0.0287	0.4402	0.0363	0.5128
CTP5	0.9893	0.0099	0.4695	0.0509	0.0117
CTP6	0.4888	0.4108	0.4208	0.0206	0.0133
CTP7	0.6458	0.4171	0.5856	0.0220	0.0222
DAS-CMOP3_9	0.3337	0.0999	0.1496	0.7515	0.4267
DAS-CMOP3_10	0.0000	0.0024	0.0795	0.0241	0.0002
DAS-CMOP3_11	0.9996	0.0137	0.1214	0.5173	0.1614
DAS-CMOP3_12	0.0000	0.0001	0.0177	0.0138	0.0065
DAS-CMOP6_9	0.3330	0.1000	0.1501	23.7690	0.3444
DAS-CMOP6_10	0.0000	0.0000	0.0000	17.5318	0.1394
DAS-CMOP6_11	1.0000	0.0000	0.0000	14.7336	0.0000
DAS-CMOP6_12	0.0000	0.0000	0.0000	22.9593	0.1433
DAS-CMOP8_9	0.1114	0.0614	0.2835	22.0217	0.6104
DAS-CMOP8_10	0.0000	0.0000	0.0000	21.0716	0.1383
DAS-CMOP8_11	1.0000	0.0000	0.0000	22.1774	0.0000
DAS-CMOP8_12	0.0000	0.0000	0.0000	21.3521	0.1358
MW1	0.0000	0.0000	0.0000	8.9253	0.5888
MW2	0.0000	0.0000	0.0000	0.6848	0.0549
MW3	0.0000	0.0000	0.0000	0.5878	0.0704
MW4	0.0000	0.0000	0.0000	4.3189	0.6268
MW5	0.0000	0.0000	0.0000	7.8080	0.3784
MW6	0.0000	0.0000	0.0000	0.6881	0.0079
MW7	0.0000	0.0000	0.0000	0.4555	0.0178
MW8	0.0000	0.0000	0.0000	0.8002	0.0080
MW9	0.0000	0.0000	0.0000	4.2789	0.0246
MW10	0.0000	0.0000	0.0000	0.7595	0.0003
MW11	0.0000	0.0000	0.0017	0.1222	0.0000
MW12	0.0000	0.0000	0.0000	4.4956	0.3126
MW13	0.0018	0.0029	0.1765	0.0587	0.0000
MW14	0.0003	0.0002	0.0391	0.1815	0.0218
Car-side impact	0.1827	0.0048	0.0284	0.1054	0.0281
Water	0.9198	0.0058	0.0612	0.0505	0.0012

In order to get a better overview of the differences between functions and function groups, each problem is shown in the FsR-$nRFB_\times$ plane, Figure 3.7(A), and in the PFd-PFcv plane, Figure 3.7(B). The FsR-$nRFB_\times$ plane confirms the initial analysis of Table 3.5. In addition, the following points are highlighted:

1. MODAct, MW, DAS-CMOP with triplets 10 and 12, CF8 and CF10 are all located near

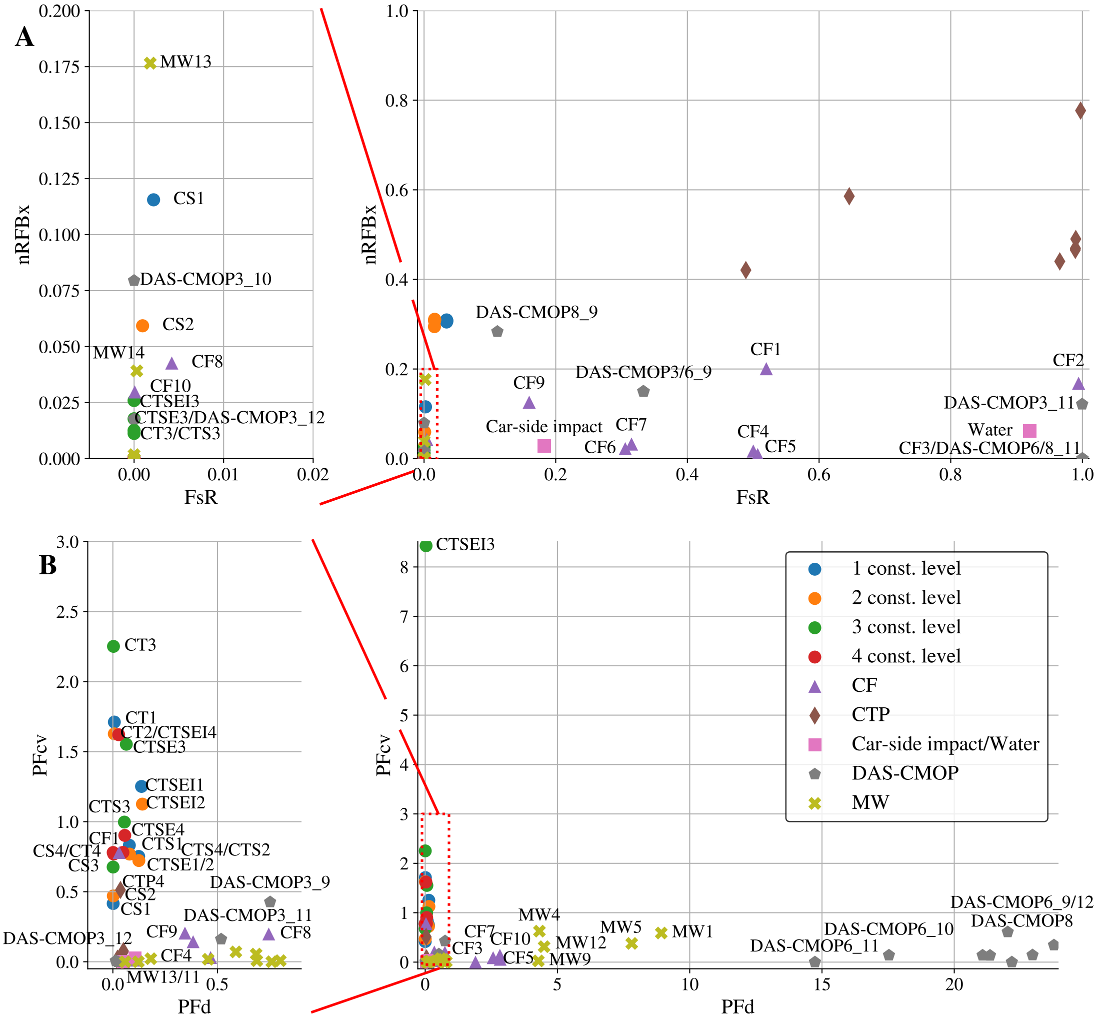


Figure 3.7: Map of the problems based on their (A) FsR-nRFB$_{\times}$ and (B) PFd-PFcv scores grouped by constraint levels or test suites

FsR $= 0$.

2. The feasibility ratio of MODAct problems is decreasing with increasing constraint level.

3. The CTP family has a unique signature with high FsR and high nRFB$_{\times}$.

Yet, there is no clear relationship between these metrics and convergence. In particular, the MW test suite is a good example that very low FsR is not directly related to convergence challenges.

Analyzing Figure 3.7(B), the newly introduced metrics exhibit the following problem clusters:

1. MODAct problems in the low PFd, high PFcv zone;
2. three groups with high PFd: (i) DAS-CMOP6 and DAS-CMOP8, (ii) MW1, MW4, MW5, MW9 and MW12 and (iii) CF3, CF5, CF7 and CF10;
3. a large group with low PFd and low PFcv, including most CTP, the car-side impact and water problems, MW11, MW13 and DAS-CMOP3_12.

Further, for the problem class CS, there is a relation between increasing PFcv values and decreasing convergence as shown in Figure 3.4. Other MODAct problems do not exhibit the same trend across constraint levels. The difficult CF functions (CF3, CF5, CF7 and CF10) have a high PFd. The opposite is true for the most difficult DAS-CMOP and MW problems. Hence, these metrics seem to capture different optimization challenges, but certainly not all of them.

In order to better understand the qualitative difference between PFd and PFcv scores and convergence, four particular functions—CT3 (low PFd, high PFcv), CF7 (high PFd, low PFcv), DAS-CMOP3_12 (low PFd, low PFcv) and MW11 (low PFd, low PFcv)—are further analyzed. Figure 3.8 shows color-maps of the minimum constraint violation values in the objective space based on the samples from $\mathcal{L}$ along with the respective best-known (CT3) or true Pareto front. It becomes apparent that PFd measures the ease of randomly sampling points in the vicinity of the Pareto front and PFcv the constraint violation of the closest known points. These particular problems represent four different situations:

1. While it is possible to find samples near the best-known Pareto front for CT3, it is harder to find feasible solutions.
2. For CF7, the sampling near the Pareto front is challenging, but it seems easier to find feasible solutions scattered all over the objective space, in particular in the direction of the Pareto front.
3. Random samples of DAS-CMOP3_12 are located in a narrowband reaching to the Pareto front and some constraints clearly guide the search process towards the optimum. Finding points outside this band is a clear challenge for diversity.
4. More uniformly distributed feasible samples near the Pareto front can be found for MW11 and here as well, some constraints clearly guide in the right direction.

The optimization challenges of the four situations are different and might thus require a different set of tools to be addressed efficiently. CF7 might, for example, be better solved by specific search operators regardless of constraints, while a specific constraint handling strategy or diversity generating operators would be needed for the others. And indeed, C-TAEA was better at solving DAS-CMOP3_12 and MW11.

Finally, an additional specificity of MODAct problems is their high number of constraints. Figure 3.9(A) represents the relative share of solutions from the random samples $\mathcal{L}$ grouped

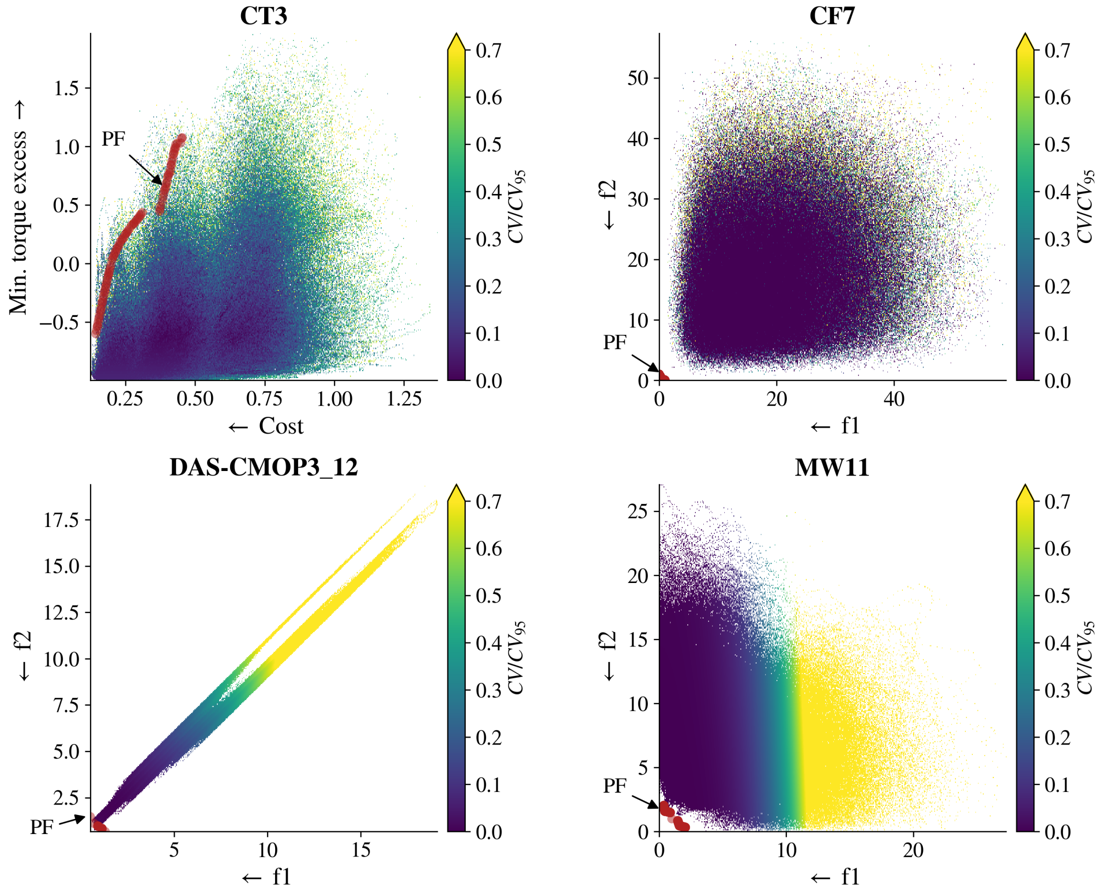


Figure 3.8: Color-maps of the normalized constraint violation values as a function of the objective values based on the random samples in $\mathcal{L}$ for CT3, CF7, DAS-CMOP3_12 and MW11 with their respective (best-known or true) Pareto front in red dots. For the generation of the color-map, the constraint violation values of points very close to each others in the objective space are aggregated using the lowest value

by number of simultaneously violated constraints for all the investigated problems with more than two constraints. The results suggest that MODAct problems are heavily constrained and have at least 50% of the search space where at least two constraints are simultaneously violated, or even at least three for the more challenging problems. For these levels, up to five constraints can be violated simultaneously in a non negligible part of the search space.

With that respect, the car-side impact problem is comparable to problems with constraint level 1. The search space of the water problem, despite its six constraints, is mostly unconstrained and when constrained, there is only one violated constraint. DAS-CMOP problems have 11 or 7 constraints most of which, however, are not simultaneously violated. The exceptions are DAS-CMOP problems with triplet 12, which have up to three simultaneous violations. MW problems have less constraints, but nonetheless, MW10 and MW11 show

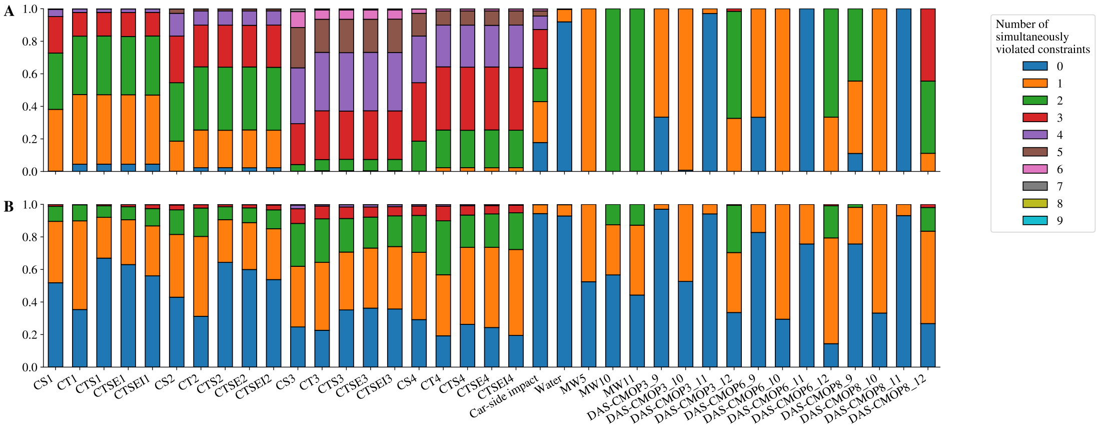


Figure 3.9: (A) Relative share of samples from $\mathcal{L}$ and (B) average share of evaluated solutions by NSGA-II or NSGA-III by number of simultaneously violated constraints for all considered problems with more than two constraints

two violated constraints throughout the samples.

This analysis is based on the whole search space and is therefore not necessarily representative of what an optimizer would encounter during its search. Figure 3.9(B) shows the average share of simultaneously violated constraints encountered during the optimizations performed by NSGA-II/III. Following the results reported by Tanabe and Oyama [173], only a small fraction of the generated solutions for the car-side impact and water problems—about 5.4% and 7.0% respectively—yield one or more simultaneously violated constraints. While the number is similar to what is shown in Figure 3.9(A) for the water problem (8.1%), the difference is significant for the car-side impact problem (80%) and suggests that in practice, not all constraints are as critical as expected.

This reduction is also observed to a more limited extent for the other problems. Yet, three and more simultaneous constraint violation have only been encountered on MODAct problems with constraint levels 3 and 4 and their share still represents 5-10%. This is a key and unique feature algorithms have to be able to deal with.

3.6 Concluding remarks

3.6.1 Key messages

In this chapter, the design problem introduced in Chapter 2 has been used to create a set of 20 constrained multiobjective optimization problems, which are representative of typical mechanical design problems. The set is built around combinations of up to five objectives and four levels of constraints, which, compared to existing test suites, puts more focus on constraints.

In order to study these problems, an extended constraint landscape analysis approach that can be applied to CMOPs has been introduced. In particular, three new metrics (nRFB$_\times$, PFd and PFcv) have been defined and complement two existing ones (FsR and RFB$_\times$). These allow a numerical investigation of the characteristics of problems in a black-box manner.

Through this approach and through a convergence study with three algorithms (NSGA-II, NSGA-III and C-TAEA), MODAct has been compared to existing benchmark problems: CTP, CF, car-side impact, water, MW and selected DAS-CMOP. The results lead to these remarks:

- The convergence study shows that the convergence quality decreases for MODAct problems as constraints are added irrespectively of the used algorithm. For the tightest constraints, getting a front near the best-known Pareto front becomes unlikely, suggesting that MODAct does cover various levels of complexity.
- In terms of algorithm selection, while C-TAEA should be preferred on many benchmark problems from literature, NSGA-II or NSGA-III perform better on the newly introduced problems.
- Many problems have a very small FsR, irrespectively of their ease of optimization.
- Different signatures in the PFd-PFcv plane have been identified and could represent different optimizations challenges: spreading the solutions along the Pareto front, sampling feasible candidates or converging to the optima. These might not all be affected by the constraints though.
- The analysis of the number of simultaneously violated constraints highlights the key role constraints in MODAct. In particular, many generated solutions violate more than one constraint.

The finding that neither NSGA-II/III and C-TAEA perform well enough on these constrained mechanical design problems is a clear motivation to further investigate constraint handling strategies taking into account severe constraints and their combination, and this will be the focus of the next chapter of this thesis. In addition, the MODAct test suite should help other researcher jump into this important topic to further improve multiobjective optimizers, in particular for mechanical design applications.

3.6.2 Creating variants of the proposed MODAct instances

The potential of MODAct exceeds the problems presented in this chapter. Once optimizers and their constraint handling strategies are capable of performing reliably on the newly introduced 20 problems, further benchmark problem can be generated. There are numerous options to do so, both within MODAct or by extending it. An easy first exploration would be the analysis of the number of stages. By looking for actuators with four or five gear stages, not only is the search space increased, but also the constraints themselves can become more complex, e.g., more gears need to fit in the same volume. Other changes can be made on the considered operating points or new combinations of objectives would result in different Pareto front shapes. Finally, constraints from the original framework could be ported to MODAct and add additional constraint levels.

Wo ist nun mein Wissen gegen dies Wirrsal?
Wo sind meine Runen gegen dies Rätsel?

Brünnhilde (Götterdämmerung, R. Wagner)

4

An Improved Constraint Handling Strategy: cEpsilon

THE formal definition of appropriate benchmark problems does not solve the previously reported core issue of performance of state-of-the-art algorithms. They do however offer a way for a standardized evaluation of existing methods and the development of new ones that may yield better performance for the type of problems of interest in this work and for mechanical design in general.

While there are algorithms specifically designed to solve constrained multiobjective optimization problems (CMOPs)—e.g., C-TAEA [105] used in Chapter 3)—most evolutionary algorithms (EAs) use so-called constraint handling strategies (CHSs) that are added to existing algorithms to handle the constraints. Typically, these CHSs change the ordering of solutions in the mating or the selection phase of EAs, but constraint-aware variation operators also exist [122].

In multiobjective optimization, algorithms need to balance between convergence (finding optimal solutions) and diversity (finding solutions with different trade-offs between objectives) [26]. In constrained multiobjective optimization, algorithms need on top to (i) find feasible regions, (ii) avoid getting trapped in non-optimal feasible regions, and (iii) maintain decision space diversity to explore the whole space.

In this chapter, existing constraint handling strategies are presented and a new method for many-constraint problems—where 'many-constraint' is defined similarly to many-objective problems: more than 3 constraints—is introduced. A total of eight methods, including the original strategy, are implemented within the framework of NSGA-III [36, 87] and evaluated on 64 benchmark problems, including 13 mechanical design problems.

4.1 Background

4.1.1 Overview

Over the past 30 years, numerous CHSs have been proposed both for single and multiobjective optimization [122]. Many researchers have suggested their own classification, e.g., [60, 105, 122, 167, 172]. In general, classifications differ depending on whether (i) the integration in the algorithm, (ii) the balance between objectives and constraints, or (iii) the numerical treatment of the constraints are considered.

There are five generic ways to integrate CHSs into algorithms [122]:

1. Penalty function methods embed the constraints within the objectives using a penalty function ϕ:
   $$\min_{\boldsymbol{x}\in\mathcal{S}} \quad \boldsymbol{f}'(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + \phi(\boldsymbol{x}) \tag{4.1}$$
   Common penalty functions include: feasibility indicator functions (also called death penalty) and linear or quadratic weighted sum of constraint violations.
2. Objectivization methods add constraints as one or several additional objectives, e.g., IDEA [149].
3. Feasibility rules alter the ranking of solutions, e.g., constrained dominance principle (CDP) [41], see Definition 3.5.
4. Special operators affect either the search operators themselves or come after the algorithm's original crossovers and mutations as 'repair' operators. They aim at easing the generation of valid solutions or repairing infeasible solutions by moving them to the feasible space [156].
5. Ensembles of techniques leverage multiple techniques to try to outperform any single approach, e.g., [116].

Penalty function and objectivization methods are often the easiest to implement and the first, in particular, is often used due to simplicity. There are however several pitfalls. Through their transformation of the fitness landscape, penalty functions can generate problems that are harder to solve. So, the effectiveness of these methods strongly depends on the appropriate selection of penalty function and weights, which is difficult [24, 152]. Adaptive methods address these issues [24, 89, 198], but most of them are not available as out-of-the-box packages. Objectivization can also significantly increase the difficulty of problems [54, 84]. Feasibility rules are also quite straight-forward to implement and in particular, CDP is at the core of many common algorithms (e.g., NSGA-II [37] and NSGA-III [36, 87]). Other feasibility rules include stochastic ranking (SR) [61, 155], a large collection of ϵ constraint-handling methods [5, 49, 167, 171], or the angle-based constrained dominance principle [52].

The latter aim at softening the 'feasibility-first' paradigm of the constrained dominance by accepting certain constraint violations: randomly in SR or using a relaxation threshold ϵ. Indeed, when feasible solutions are systematically preferred over infeasible solutions, algorithms tend to be pushed to feasible areas faster, but can get trapped in non-optimal areas. Further, it is not uncommon for optimal solutions to lie on the boundaries of feasible space—in particular for mechanical system, it is expected that at least some constraints are so-called active. By systematically discarding infeasible solutions, their information is lost. On the contrary, ϵ-constrained handling methods preserve solutions within ϵ of the feasible space, and by progressively reducing the gap, hope to more easily find solutions lying close to the feasibility border. These different approaches lead to a second classification:

1. feasibility-driven: feasible solutions are always preferred;
2. optimality-driven: better solutions are always preferred;
3. balanced: objectives and constraints are both considered.

In addition, algorithms can follow fixed or dynamic strategies. NSGA-II or NSGA-III use for example a fixed feasibility-driven approach, while the push and pull framework [53] has an optimality-driven phase, followed by a balanced phase and finally a feasibility-driven phase. The ϵ-constrained methods follow a balanced first and then feasibility-driven phase.

Finally, there are multiple ways for constraints to be numerically treated by an algorithm—i.e., the measure of the 'amount' of violation:

1. the constraint violation CV, see Eq. (3.1);
2. the number of active constraints;
3. the rank of the solutions in terms of constraints;
4. combinations of the above.

While the use of the CV value is most common, alternate approaches exist [24]. Solely relying on the CV can indeed induce numerical biases if multiple constraints with different scales are combined. The multiple constraint ranking method [60] relies for example on ranking the solutions in terms of constraints and using the obtained rank as the fitness. Others proposed alternate CV definitions, e.g., Asafuddoula et al. [5] suggest the use of CV' which combines CV and number of active constraints:

$$CV'(\boldsymbol{x}) = m_1 \sum_{j=1}^{p} \langle g_j(\boldsymbol{x}) \rangle + m_2 \sum_{k=1}^{q} |h_k(\boldsymbol{x})| \tag{4.2}$$

where m_1 and m_2 are the number of active inequality and equality constraints.

While CDP remains the most used strategy today, it is not due to a lack of alternatives, but rather thanks to its parameterless nature and the lack of readily available implementations and systematical performance review also on applied optimization problems.

4.1.2 Selected ϵ constraint-handling methods

Following the observed challenges of CDP in Chapter 3, more balanced approaches are sought to solve MODAct problems. In particular, ϵ-constrained methods are thought to be effective for highly constrained problems [50], since they introduce a relaxation scheme. ϵ-constrained methods are also common in operational research—e.g., in NOMADS [6, 101] where it is called barrier method. In this section, selected ϵ-constrained methods for EA are detailed.

Takahama and Sakai [171] introduced the original ϵ-constrained method (E) by modifying CDP to have some tolerance ϵ:

Definition 4.1. Given two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and for any $\epsilon \geq 0$, $\boldsymbol{x}_1$ is said to ϵ-constrained-dominate $\boldsymbol{x}_2$, written $\boldsymbol{x}_1 \preceq_\epsilon \boldsymbol{x}_2$, if:

$$\boldsymbol{x}_1 \preceq_\epsilon \boldsymbol{x}_2 \Leftrightarrow \begin{cases} \boldsymbol{x}_1 \preceq \boldsymbol{x}_2 & \text{if } CV(\boldsymbol{x}_1), CV(\boldsymbol{x}_2) \leq \epsilon \\ \boldsymbol{x}_1 \preceq \boldsymbol{x}_2 & \text{if } CV(\boldsymbol{x}_1) = CV(\boldsymbol{x}_2) \\ CV(\boldsymbol{x}_1) < CV(\boldsymbol{x}_2) & \text{otherwise} \end{cases} \tag{4.3}$$

where $\preceq$ is the Pareto dominance operator, recall Definition 1.1.

Definition 4.1 can be seen as a generalization of CDP, since when $\epsilon = 0$, they are equivalent[1]. In addition, Takahama and Sakai [171] suggest that ϵ should be updated at each generation using the following scheme.

$$\epsilon(0) = CV(\boldsymbol{x}_\theta) \tag{4.4}$$

$$\epsilon(t) = \begin{cases} \epsilon(0)\left(1 - \frac{t}{T_c}\right)^{c_p} & \text{if } 0 < t < T_c \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

where $\boldsymbol{x}_\theta$ is the top θ^{th} individual of the initial population, $c_p \in [2, 10]$ a parameter to control the reduction rate, T_c the cut-off generation after which $\epsilon = 0$ and t the current generation. θ represents the number of solutions considered valid initially, and is set as a fraction of the population size μ of the used algorithm: $\theta = \lfloor \gamma\mu \rfloor$. Similarly, T_c is set as a fraction of the total number of generations T_{max}: $T_c = \lambda T_{\text{max}}$.

With this scheme, ϵ is strictly decreasing until it reaches 0 at T_c. Its general shape and the effects of c_p and T_c are illustrated in Figure 4.1. Its initial value depends on the initial population and θ allows to select the proportion of the population that should be considered 'feasible'.

---

[1] There is a small difference for two infeasible solutions with the same constraint violation level.

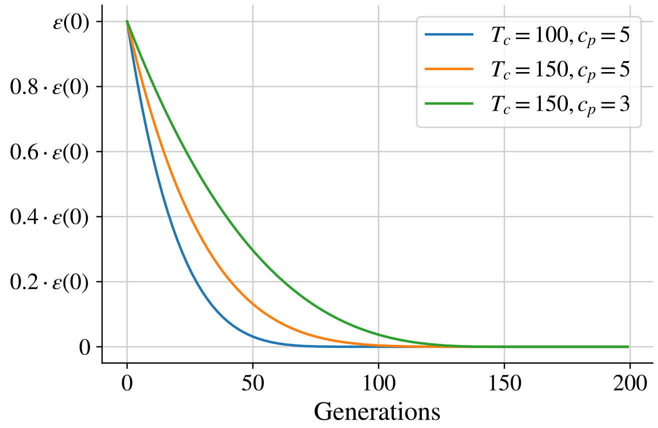


Figure 4.1: Evolution of ϵ along generations for different parameter values.

The challenges of this method are the high number of parameters, which need to be adapted for each problem and the strict decay, which could trap the algorithm in sub-optimal feasible area. To address these shortcomings, Asafuddoula et al. [5] proposed the adaptive epsilon method (*aE*). It uses the same dominance relation (4.3), but the acceptable violation threshold is set based on the feasability ratio (FsR)—see Definition 3.1—of the current population and its average CV':

$$\epsilon = CV'_{\text{mean}}\text{FsR} \tag{4.6}$$

That way the threshold is set to 0 for either totally feasible ($CV' = 0$) or infeasible populations ($\text{FsR} = 0$). In between, FsR adjusts the part of the population with acceptable constraint violation.

Fan et al. [49] proposed their own improvement of the original method that includes a mechanism to dynamically increase the value of ϵ for given situations. Their improved epsilon method (*iE*) uses the same dominance relation (4.3), but the ϵ update scheme is changed:

$$\epsilon(0) = CV(\boldsymbol{x}_\theta) \tag{4.7}$$

$$\epsilon(t) = \begin{cases} \epsilon(0)(1 - \frac{t}{T_c}) & \text{if FsR} < \alpha \text{ and } 0 < t < T_c \\ (1 + \tau)CV_{\max} & \text{if FsR} \geq \alpha \text{ and } 0 < t < T_c \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

where $CV_{\max}$ is the maximum CV value seen during an optimization, and $\tau, \alpha \in [0, 1]$ are two additional parameters that control when and by how much ϵ increases.

In 2020, Stanovov et al. [167] proposed five new variants, tested on single-objective optimization problems so far. Two of these variants can be used in a multiobjective setting. These methods are 'population-based': the ϵ value is set at each generation based on the CV values of the current population and instead, the parameter θ is decreased iteratively. Their first method (pE) still keeps the dominance relation (4.3), but changes the update of ϵ to:

$$\theta(t) = \left\lfloor \gamma\mu\left(1 - \frac{t}{T_c}\right)^{c_p} \right\rfloor \tag{4.9}$$

$$\epsilon(t) = \begin{cases} CV(\boldsymbol{x}_{\theta(t)}) & \text{if } t < T_c \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

Their second method (ipE) introduces a per-constraint ϵ_j value that is controlled similarly to pE, assuming that the optimization problem only has inequality constraints[2].

$$\epsilon_j(t) = \begin{cases} g_j\left(\boldsymbol{x}_{\theta(t)}\right) & \text{if } t < T_c \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

Moving to a per-constraint view affects the dominance relation since the individual constraints g_j need now to be compared against their respective ϵ_j.

$$\boldsymbol{x}_1 \preceq_{\epsilon,ip} \boldsymbol{x}_2 \Leftrightarrow \begin{cases} \boldsymbol{x}_1 \preceq \boldsymbol{x}_2 & \text{if } CV^{\epsilon}(\boldsymbol{x}_1), CV^{\epsilon}(\boldsymbol{x}_2) \leq \sum_{j=1}^{p} \epsilon_j \\ \boldsymbol{x}_1 \preceq \boldsymbol{x}_2 & \text{if } CV^{\epsilon}(\boldsymbol{x}_1) = CV^{\epsilon}(\boldsymbol{x}_2) \\ CV^{\epsilon}(\boldsymbol{x}_1) < CV^{\epsilon}(\boldsymbol{x}_2) & \text{otherwise} \end{cases} \tag{4.12}$$

where CV^{ϵ} is calculated using these equations:

$$\phi_j^{\epsilon}(\boldsymbol{x}) = \begin{cases} g_j(\boldsymbol{x}) & \text{if } g_j(\boldsymbol{x}) > \epsilon_j \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

$$CV^{\epsilon}(\boldsymbol{x}) = \sum_{j=1}^{p} \phi_j^{\epsilon}(\boldsymbol{x}) \tag{4.14}$$

With this approach, Stanovov et al. [167] make a first step to consider inequality constraints individually, but they keep some sort of 'group-level' ϵ formed by the sum of all individual ϵ_j.

---

[2] In practice, this is not a strong assumption since strict equality constraints are difficult to implement numerically and are thus transformed into inequality constraints: $|h_k(\boldsymbol{x})| - \delta \leq 0$, where δ is a small number.

4.2 Adapted per-constraint epsilon values

Per-constraint relaxation has existed for many years in non-stochastic optimizers [6, 101]. The concept itself is, however, of great interest in the case of problems with many constraints. Indeed, when CMOPs have multiple simultaneously active constraints—and of potentially different scales—the *CV* value can become blurry. A reduction of the CV value could actually hide increased violations of other constraints. Thus, the ϵ-constrained methods acting on the *CV* cannot guarantee that all constraints effectively improve at each iteration.

A new strategy called cEpsilon (*cE*) is introduced here to address this shortcoming. The update scheme for ϵ_j is a per-constraint transposition of the original ϵ-constrained method.

$$\epsilon_j(0) = g_j(\boldsymbol{x}_\theta) \tag{4.15}$$

$$\epsilon_j(t) = \begin{cases} \epsilon_j(0)\left(1 - \frac{t}{T_c}\right)^{c_p} & \text{if } 0 < t < T_c \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

Using the same CV^ϵ as ipE, the preference relation is, however, adapted to be stricter:

$$\boldsymbol{x}_1 \preceq_{\epsilon,c} \boldsymbol{x}_2 \Leftrightarrow \begin{cases} \boldsymbol{x}_1 \preceq \boldsymbol{x}_2 & \text{if } CV^\epsilon(\boldsymbol{x}_1), CV^\epsilon(\boldsymbol{x}_2) = 0 \\ \boldsymbol{x}_1 \preceq \boldsymbol{x}_2 & \text{if } CV^\epsilon(\boldsymbol{x}_1) = CV^\epsilon(\boldsymbol{x}_2) \\ CV^\epsilon(\boldsymbol{x}_1) < CV^\epsilon(\boldsymbol{x}_2) & \text{otherwise} \end{cases} \tag{4.17}$$

The proposed method guarantees that at each step the thresholds for all constraints are lowered and the requirements are effectively tightened. Thus, this method generates a relatively high selection pressure towards finding good feasible regions, while offering some tolerance adjusted to the scale of each constraint. In addition, it is possible to have per-constraint θ, c_p and T_c parameters, allowing a very fine control over each individual constraint. The downside, however, is the steep increase in the number of parameters. So, unless domain specific knowledge is available that can guide the selection of the parameters, this approach remains quite theoretical.

4.3 Methods of the numerical investigations

A large-scale investigation is conducted to evaluate the performance of the ϵ-constrained methods previously presented along with CDP and SR. To limit the impact of additional effects, a single base algorithm is chosen and only the CHS is changed. NSGA-III [36, 87] is selected. NSGA-III is a reference-direction based many-objective optimizer. In the context of this work, it offers two important advantages:

1. NSGA-III allows a parallel evaluation of the objective and constraint functions, which

is important in the context of ‘real-world’ benchmark problems such as MODAct.

2. Decomposition-based algorithms struggle with certain Pareto front shapes [85], which, from the performance of C-TAEA, seems to apply to MODAct problems.

This section clarifies the integration of the considered CHSs and details the methods used for the numerical investigation.

4.3.1 Integration within NSGA-III

The handling of constraints by NSGA-III is described by Jain and Deb [87]. Compared to the algorithm for unconstrained problems, two parts are adapted:

1. Modification of the non-dominated sorting procedure to use CDP, effectively sorting the population by *CV* value and additionally ranking feasible solutions into non-dominated fronts.
2. Modification of parent selection procedure, replacing the random parent selection by a binary tournament preferring less constrained solutions.

This forms the original CHS for NSGA-III and is referred to as the CDP method. The implementation of NSGA-III within pymoo [9] is used and the additional *SR*, *E*, *iE*, *pE*, *ipE*, and *cE* methods have been implemented to fit into the same frame. Only an initialization and an update steps have been added for the ϵ-constrained methods, see Algorithm 4.1.

Algorithm 4.1 Overview of NSGA-III with an ϵ update scheme

```
$P_0 \leftarrow$ InitializePopulation
InitializeCHS($P_0$)
$t \leftarrow 0$
while $t < T_{\max}$ do
    $M_t \leftarrow$ SelectParents($P_t$)
    $Q_t \leftarrow$ Variate($M_t$)
    $R_t \leftarrow P_t \cup Q_t$
    $P_{t+1} \leftarrow$ Select($R_t$)
    UpdateCHS($P_{t+1}$)
    $t \leftarrow t + 1$
return $P_t$
```

Consequently, the non-dominated sorting procedure is changed to follow the preference relation of each method. The binary tournament is also adapted to use the appropriate preference relation to discriminate between “feasible” and “infeasible” solutions, followed by random selection if both parents fit in the same category.

Finally, a stochastic ranking *SR* method is also implemented. In the *SR* method, infeasible solutions have a probability p_f to be treated as feasible solutions [61, 155]. In practice, this is achieved by randomly setting the CV value of infeasible solutions to 0 with a probability of p_f and then applying the original procedure of NSGA-III. In the binary tournament, this is repeated at every tournament, while it is done only once prior to the normal non-dominated sorting. Thus, this implementation slightly deviates from the original 'bubble-sort' procedure [155], to benefit from the fast sorting algorithm.

4.3.2 Experimental conditions and data analysis

Building on the results from Chapter 3, the following benchmark problems are selected: C3DTLZ4 [87], DAS-CMOP with complexity level 9, 10, 11, and 12 [51], and MW [112]. In addition, 12 problems from MODAct as well as the windturbine design problem proposed at the 3$^{\text{rd}}$ Evolutionary Computation Competition by the Japanese Society of Evolutionary Computation [91] are included. In total, the various CHSs will be tested on 64 problems with different properties, summarized in Table 4.1, which also indicates the number of optimization generations $T_{\max}$ performed for each problem. $T_{\max}$ has been adapted to the complexity of the various problems. The only exception is the Windturbine problem where each objective and constraint evaluation takes about 3 s and the number of evaluations had to be restricted due to time considerations[3].

Each optimization (pair of problem and CHS) is executed 30 times. The parameters for NSGA-III are specified in Table 4.2 and follow common practice for this algorithm. The reference directions are generated using Das and Dennis' approach [33]. As recommended by Brockhoff et al. [12], an unbounded external archive (UEA) is used to collect all feasible non-dominated solutions generated through an optimization run.

The hypervolume indicator [204] is used to evaluate the convergence and diversity of the obtained archive of each run, and is calculated by the implementations of the Walking-Fish Group [28, 189][4]. The best-known Pareto fronts of each problem, as provided by their authors or obtained by aggregating all the solutions from the UEAs, are used to estimate the ideal $\boldsymbol{z}^*$ and nadir $\boldsymbol{z}^{\text{nad}}$ points. They serve then to normalize the obtained fronts and calculate the hypervolume with a common reference point $\boldsymbol{r} = (1.1, \dots, 1.1)^T$. In this context, where no cross problem comparison is required, there is no need to go all the way to the relative hypervolume error, Eq. (3.10).

For the statistical analysis of the difference between the CHSs, the non-parametric Kruskal-Wallis test [98] is applied with the null hypothesis that all CHSs perform equally well on a given problem with a confidence interval of 99%. If the null hypothesis is rejected, Mann-Whitney U tests [119] with Holm's correction for family-wise error control [73] are applied with a confidence interval of 99% to perform the multiple pairwise comparisons. The statistics are calculated with pingouin [182].

---

[3] With a total budget of 42 000 per optimization, this experiment already greatly exceeds the budget of 10 000 of the competition.

[4] A python interface has been developed and is available at: https://github.com/epfl-lamd/hvwfg

Table 4.1: Number of objectives, decision variables and constraints of the selected benchmark CMOPs and the defined number of generations for each problem.

	m	n	p	q	$T_{\max}$
C3DTLZ4	3	12	3	0	750
CS1, CT1	2	20	7	0	1500
CS2, CT2	2	20	8	0	1500
CS3, CT3	2	20	10	0	1500
CS4, CT4	2	20	9	0	1500
CTS1	3	20	7	0	1500
CTS2	3	20	8	0	1500
CTS3	3	20	10	0	1500
CTS4	3	20	9	0	1500
MW1, MW2, MW6, MW9	2	15	1	0	600
MW3, MW7, MW12, MW13	2	15	2	0	600
MW4, MW8, MW14	3	15	1	0	600
MW5, MW10	2	15	3	0	600
MW11	2	15	4	0	600
DAS-CMOP{1..6}_9	2	30	11	0	1500
DAS-CMOP{1..6}_10	2	30	11	0	1500
DAS-CMOP{1..6}_11	2	30	11	0	1500
DAS-CMOP{1..6}_12	2	30	11	0	1500
DAS-CMOP{7..9}_9	3	30	7	0	1500
DAS-CMOP{7..9}_10	3	30	7	0	1500
DAS-CMOP{7..9}_11	3	30	7	0	1500
DAS-CMOP{7..9}_12	3	30	7	0	1500
Windturbine	5	32	22	0	200

Table 4.2: Parameters used to configure each run of NSGA-III.

Parameter	NSGA-III
Population size μ	210
Number of function (i.e., solution) evaluations	$\mu \cdot T_{\max}$
Mutation η_m	20
Mutation rate	$1/n$
Crossover η_c	20
Crossover probability (CXPB)	0.9

Table 4.3: Parameters and their range for the parameter tuning configuration of irace.

Parameter	Accepted value	Condition
CHS	$\{E, iE, pE, ipE, cE, SR\}$	-
p_f	$[0, 1]$	Only for *SR*
λ	$[0, 1]$	Except for *SR*
c_p	$[2, 10]$	Except for *SR*
γ	$[0, 1]$	Except for *SR*
α	$[0, 1]$	Only for *iE*
τ	$[0, 1]$	Only for *iE*

4.3.3 Parameter tuning setup

Since most considered CHSs have many parameters, a parameter tuning step—also called automatic algorithm configuration or meta-optimization—is performed using irace [110]. Irace is configured to search for the CHS and its parameters that maximize the hypervolume of the obtained Pareto front following the procedure suggested for multiobjective evolutionary algorithms (MOEAs) [108]. The acceptable parameter values are summarized in Table 4.3. To consider only the effects of the CHS, only those parameters are considered and the parameters of NSGA-III are excluded. The tuning procedure is executed repeatedly for a maximum of 5000 experiments per run and for the different problem families—MODAct, MW and DAS-CMOP grouped by complexity level. Due to time limitations, the Windturbine problem has not been used for tuning.

4.4 Results

4.4.1 Automated tuning of parameters

The best and second-best configuration of each run of irace have been collected. Similar configurations were manually aggregated. In the end, a total of 10 configurations were selected for the six methods with parameters. Including the two parameterless methods (CDP and *aE*), 12 configurations are to be evaluated, see Table 4.4. Interestingly, the tuning process suggested the stochastic ranking method with $p_f = 1$ (SR_2), which is equivalent to solving the unconstrained problem, despite the fact that only feasible solutions contribute to the hypervolume.

4.4.2 Performance comparison

With 12 configurations and 64 problems, a total of 23 040 optimizations have been performed using the school's high power computing clusters. The median achieved hypervolume and the interquartile range (IQR) for each configuration on each problem is reported

Table 4.4: List of the considered CHSs with the parameters of the various configurations obtained through the tuning procedure.

Name	Symbol	Parameters
Constrained dominance principle	CDP	–
Adaptive epsilon	aE	–
Stochastic ranking	SR_1	$p_f = 0.36$
	SR_2	$p_f = 1$
Epsilon	E_1	$\lambda = 0.18, c_p = 8.76, \gamma = 0.72$
Improved epsilon	iE_1	$\lambda = 0.26, c_p = 9.51, \gamma = 0.66, \alpha = 0.52, \tau = 0.8$
	iE_2	$\lambda = 0.4, c_p = 8.11, \gamma = 0.6, \alpha = 0.57, \tau = 0.66$
Population epsilon	pE_1	$\lambda = 0.73, c_p = 6.96, \gamma = 0.87$
	pE_2	$\lambda = 0.57, c_p = 7.55, \gamma = 0.38$
Individual population epsilon	ipE_1	$\lambda = 0.93, c_p = 3.19, \gamma = 0.23$
Per-constraint Epsilon	cE_1	$\lambda = 0.11, c_p = 9.11, \gamma = 0.66$
	cE_2	$\lambda = 0.5, c_p = 4.52, \gamma = 0.59$

in Table 4.5. Without statistical considerations, the following trends are visible:

1. CDP seems best for the Windturbine design problem;
2. cE seems most interesting for MODAct problems;
3. iE performs best on C3DTLZ4 and MW problems;
4. SR seems most suitable for DAS-CMOP problems.

In particular, the unconstrained optimizer SR_2 seems to be the most suitable method for DAS-CMOP with complexity level 11, while it fails to find feasible and interesting solutions for most optimization runs on the Windturbine and MODAct problems. There are few other singularities:

1. None of the 360 optimizations has been able to find any feasible solutions to DAS-CMOP1_12.
2. Some methods have not been able to find any feasible solutions for problems CS3, DAS-CMOP3_12, and DAS-CMOP6_12.
3. Excluding DAS-CMOP1_12, ipE and cE are the only methods that have found solutions for all other problems.

The numerical results, however, also suggest that multiple methods have similar performance. The results of the statistical analysis are reported in Table 4.6 by indicating how many times a given strategy was better/equal/worse than others for each problem. For eight problems—namely DAS-CMOP1_11, DAS-CMOP1_12, DAS-CMOP4_9, DAS-CMOP5_9,

Table 4.5: Median and interquartile range (IQR) of the hypervolume for the obtained Pareto fronts for each problem and CHS configuration. The cell with the highest median value per problem is highlighted in gray.

	CDP	aE	E_1	cE_1	cE_2	ipE_1	iE_1	iE_2	pE_1	pE_2	SR_1	SR_2
C3DTLZ4	1.203 (5.858×10^{-5})	1.203 (6.166×10^{-5})	1.203 (7.824×10^{-5})	1.203 (6.540×10^{-5})	1.203 (6.431×10^{-5})	1.203 (6.022×10^{-5})	1.203 (1.078×10^{-4})	**1.203** **(1.456×10^{-4})**	1.203 (1.010×10^{-4})	1.203 (4.386×10^{-5})	1.198 (2.525×10^{-3})	1.097 (1.145×10^{-1})
CS1	9.305×10^{-1} (7.214×10^{-2})	9.427×10^{-1} (1.733×10^{-2})	9.467×10^{-1} (6.755×10^{-2})	9.435×10^{-1} (3.307×10^{-2})	9.243×10^{-1} (3.335×10^{-2})	9.359×10^{-1} (1.359×10^{-1})	9.342×10^{-1} (3.078×10^{-2})	9.408×10^{-1} (3.601×10^{-2})	**9.588×10^{-1}** **(1.845×10^{-2})**	9.325×10^{-1} (1.335×10^{-1})	8.922×10^{-1} (1.167×10^{-1})	5.323×10^{-2} (8.870×10^{-2})
CS2	9.030×10^{-1} (1.411×10^{-1})	9.007×10^{-1} (1.553×10^{-1})	9.199×10^{-1} (5.067×10^{-2})	9.096×10^{-1} (5.313×10^{-2})	9.105×10^{-1} (3.766×10^{-2})	9.016×10^{-1} (1.514×10^{-1})	9.252×10^{-1} (5.262×10^{-2})	9.037×10^{-1} (6.638×10^{-2})	**9.283×10^{-1}** **(4.843×10^{-2})**	9.038×10^{-1} (1.462×10^{-1})	8.355×10^{-1} (1.341×10^{-1})	0.000 (5.250×10^{-2})
CS3	5.096×10^{-1} (4.261×10^{-1})	4.912×10^{-1} (3.853×10^{-1})	0.000 (0.000)	5.241×10^{-1} (1.962×10^{-1})	**7.197×10^{-1}** **(3.142×10^{-1})**	3.639×10^{-1} (4.586×10^{-1})	0.000 (0.000)	0.000 (0.000)	7.009×10^{-1} (1.767×10^{-1})	4.219×10^{-1} (4.880×10^{-1})	3.890×10^{-1} (2.959×10^{-1})	0.000 (0.000)
CS4	3.813×10^{-1} (1.333×10^{-1})	4.008×10^{-1} (1.800×10^{-1})	3.129×10^{-1} (1.187×10^{-1})	7.565×10^{-1} (1.227×10^{-1})	**8.896×10^{-1}** **(8.444×10^{-2})**	3.643×10^{-1} (2.339×10^{-1})	3.648×10^{-1} (6.064×10^{-2})	3.931×10^{-1} (3.463×10^{-2})	1.516×10^{-1} (2.501×10^{-1})	4.155×10^{-1} (1.544×10^{-1})	4.219×10^{-1} (1.563×10^{-1})	0.000 (0.000)
CT1	8.931×10^{-1} (1.433×10^{-2})	8.940×10^{-1} (1.124×10^{-2})	8.873×10^{-1} (3.545×10^{-2})	8.928×10^{-1} (2.881×10^{-2})	8.844×10^{-1} (2.401×10^{-2})	**8.961×10^{-1}** **(8.039×10^{-3})**	8.701×10^{-1} (6.789×10^{-2})	8.743×10^{-1} (6.859×10^{-2})	8.837×10^{-1} (7.015×10^{-2})	8.948×10^{-1} (9.535×10^{-3})	8.926×10^{-1} (9.928×10^{-3})	3.087×10^{-1} (9.833×10^{-2})
CT2	8.906×10^{-1} (2.269×10^{-2})	8.914×10^{-1} (2.155×10^{-2})	8.915×10^{-1} (6.240×10^{-2})	8.915×10^{-1} (1.608×10^{-2})	8.787×10^{-1} (3.141×10^{-2})	8.891×10^{-1} (2.259×10^{-2})	8.895×10^{-1} (5.591×10^{-2})	8.354×10^{-1} (6.650×10^{-2})	8.904×10^{-1} (7.094×10^{-2})	**8.968×10^{-1}** **(1.440×10^{-2})**	8.931×10^{-1} (9.510×10^{-3})	2.546×10^{-1} (7.936×10^{-2})
CT3	7.223×10^{-1} (1.004×10^{-1})	**7.265×10^{-1}** **(9.967×10^{-2})**	5.493×10^{-1} (4.613×10^{-2})	6.574×10^{-1} (1.283×10^{-1})	6.827×10^{-1} (1.707×10^{-1})	6.546×10^{-1} (1.682×10^{-1})	5.579×10^{-1} (4.254×10^{-2})	5.347×10^{-1} (5.602×10^{-2})	5.837×10^{-1} (5.962×10^{-2})	7.094×10^{-1} (1.242×10^{-1})	6.450×10^{-1} (1.462×10^{-1})	0.000 (0.000)
CT4	4.953×10^{-1} (1.268×10^{-1})	5.199×10^{-1} (1.193×10^{-1})	1.275×10^{-1} (3.589×10^{-2})	7.297×10^{-1} (1.457×10^{-1})	**7.836×10^{-1}** **(2.339×10^{-2})**	4.909×10^{-1} (1.300×10^{-1})	1.253×10^{-1} (1.732×10^{-2})	1.176×10^{-1} (1.265×10^{-1})	1.841×10^{-1} (1.212×10^{-1})	4.604×10^{-1} (1.153×10^{-1})	4.177×10^{-1} (9.335×10^{-2})	0.000 (0.000)
CTS1	3.785×10^{-1} (2.136×10^{-3})	3.792×10^{-1} (1.851×10^{-3})	3.625×10^{-1} (1.134×10^{-2})	**3.794×10^{-1}** **(1.660×10^{-3})**	3.791×10^{-1} (3.554×10^{-3})	3.790×10^{-1} (2.279×10^{-3})	3.523×10^{-1} (2.482×10^{-2})	3.524×10^{-1} (1.164×10^{-2})	3.659×10^{-1} (8.176×10^{-3})	3.760×10^{-1} (5.343×10^{-3})	3.683×10^{-1} (6.179×10^{-3})	3.179×10^{-1} (8.455×10^{-3})
CTS2	3.742×10^{-1} (3.359×10^{-3})	3.764×10^{-1} (4.270×10^{-3})	3.588×10^{-1} (1.005×10^{-2})	**3.764×10^{-1}** **(2.044×10^{-3})**	3.763×10^{-1} (2.055×10^{-3})	3.744×10^{-1} (2.935×10^{-3})	3.422×10^{-1} (1.603×10^{-2})	3.556×10^{-1} (1.192×10^{-2})	3.647×10^{-1} (8.506×10^{-3})	3.729×10^{-1} (5.163×10^{-3})	3.646×10^{-1} (7.282×10^{-3})	2.969×10^{-1} (1.221×10^{-2})
CTS3	2.951×10^{-1} (9.029×10^{-2})	2.944×10^{-1} (9.313×10^{-2})	2.617×10^{-1} (5.524×10^{-2})	3.152×10^{-1} (2.911×10^{-2})	**3.445×10^{-1}** **(4.269×10^{-2})**	2.839×10^{-1} (1.170×10^{-1})	2.766×10^{-1} (4.317×10^{-2})	2.585×10^{-1} (5.043×10^{-2})	2.518×10^{-1} (6.070×10^{-2})	2.439×10^{-1} (5.022×10^{-2})	2.326×10^{-1} (8.790×10^{-2})	0.000 (0.000)
CTS4	2.063×10^{-1} (1.067×10^{-1})	2.088×10^{-1} (7.939×10^{-2})	3.136×10^{-1} (1.341×10^{-2})	3.211×10^{-1} (1.944×10^{-2})	**3.548×10^{-1}** **(9.202×10^{-3})**	1.490×10^{-1} (8.838×10^{-2})	3.194×10^{-1} (1.048×10^{-2})	3.201×10^{-1} (1.058×10^{-2})	2.548×10^{-1} (7.186×10^{-2})	1.770×10^{-1} (9.457×10^{-2})	2.005×10^{-1} (1.065×10^{-1})	0.000 (0.000)
DAS-CMOP1_9	4.531×10^{-1} (1.281×10^{-2})	4.553×10^{-1} (5.587×10^{-3})	4.514×10^{-1} (2.094×10^{-2})	4.518×10^{-1} (1.438×10^{-2})	4.463×10^{-1} (2.278×10^{-2})	4.531×10^{-1} (1.281×10^{-2})	4.424×10^{-1} (2.098×10^{-2})	4.492×10^{-1} (2.540×10^{-2})	4.553×10^{-1} (6.926×10^{-3})	**4.585×10^{-1}** **(4.232×10^{-2})**	4.569×10^{-1} (1.559×10^{-2})	4.243×10^{-1} (4.589×10^{-2})
DAS-CMOP1_10	4.592×10^{-1} (4.367×10^{-2})	4.577×10^{-1} (4.743×10^{-2})	4.723×10^{-1} (2.658×10^{-2})	4.747×10^{-1} (4.538×10^{-2})	4.698×10^{-1} (2.824×10^{-2})	4.623×10^{-1} (4.997×10^{-2})	4.584×10^{-1} (5.318×10^{-2})	4.451×10^{-1} (4.490×10^{-2})	4.692×10^{-1} (3.680×10^{-2})	4.637×10^{-1} (3.748×10^{-2})	**4.975×10^{-1}** **(3.363×10^{-2})**	4.526×10^{-1} (2.779×10^{-2})
DAS-CMOP1_11	7.453×10^{-1} (5.926×10^{-2})	7.453×10^{-1} (5.926×10^{-2})	7.453×10^{-1} (5.926×10^{-2})	7.453×10^{-1} (5.926×10^{-2})	7.453×10^{-1} (5.926×10^{-2})	7.453×10^{-1} (5.926×10^{-2})	7.485×10^{-1} (7.313×10^{-2})	7.418×10^{-1} (6.955×10^{-2})	7.453×10^{-1} (5.926×10^{-2})	7.453×10^{-1} (5.926×10^{-2})	7.430×10^{-1} (4.721×10^{-2})	**7.874×10^{-1}** **(7.837×10^{-2})**
DAS-CMOP1_12	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
DAS-CMOP2_9	5.469×10^{-1} (7.053×10^{-3})	5.469×10^{-1} (9.613×10^{-3})	**5.475×10^{-1}** **(2.086×10^{-2})**	5.470×10^{-1} (1.204×10^{-2})	5.420×10^{-1} (1.307×10^{-2})	5.469×10^{-1} (7.053×10^{-3})	5.433×10^{-1} (1.716×10^{-2})	5.452×10^{-1} (1.520×10^{-2})	5.451×10^{-1} (4.121×10^{-2})	5.469×10^{-1} (7.399×10^{-3})	5.456×10^{-1} (3.912×10^{-2})	5.122×10^{-1} (6.709×10^{-2})
DAS-CMOP2_10	6.415×10^{-1} (2.846×10^{-2})	6.376×10^{-1} (2.926×10^{-2})	6.428×10^{-1} (1.193×10^{-2})	6.564×10^{-1} (1.607×10^{-2})	6.449×10^{-1} (9.147×10^{-3})	6.415×10^{-1} (2.589×10^{-2})	6.419×10^{-1} (1.667×10^{-2})	6.441×10^{-1} (1.563×10^{-2})	6.447×10^{-1} (2.286×10^{-2})	6.380×10^{-1} (1.708×10^{-2})	**6.614×10^{-1}** **(2.797×10^{-2})**	6.370×10^{-1} (7.962×10^{-3})
DAS-CMOP2_11	9.849×10^{-1} (8.771×10^{-1})	9.849×10^{-1} (8.771×10^{-1})	9.849×10^{-1} (8.771×10^{-1})	9.849×10^{-1} (8.771×10^{-1})	9.849×10^{-1} (8.771×10^{-1})	9.849×10^{-1} (8.771×10^{-1})	9.874×10^{-1} (6.612×10^{-1})	9.865×10^{-1} (8.774×10^{-1})	9.849×10^{-1} (8.771×10^{-1})	9.849×10^{-1} (8.771×10^{-1})	9.845×10^{-1} (2.537×10^{-2})	**1.012** **(9.914×10^{-3})**
DAS-CMOP2_12	5.500×10^{-1} (1.192×10^{-1})	4.931×10^{-1} (8.661×10^{-2})	6.092×10^{-1} (1.137×10^{-1})	5.876×10^{-1} (1.340×10^{-1})	5.845×10^{-1} (1.455×10^{-1})	5.064×10^{-1} (1.243×10^{-1})	6.268×10^{-1} (7.762×10^{-2})	**6.330×10^{-1}** **(4.154×10^{-2})**	5.198×10^{-1} (1.365×10^{-1})	4.926×10^{-1} (1.277×10^{-1})	5.077×10^{-1} (9.595×10^{-2})	6.039×10^{-1} (1.228×10^{-1})
DAS-CMOP3_9	**5.946×10^{-1}** **(3.321×10^{-2})**	5.748×10^{-1} (4.643×10^{-2})	5.429×10^{-1} (1.303×10^{-1})	5.532×10^{-1} (6.617×10^{-2})	4.115×10^{-1} (7.197×10^{-2})	5.946×10^{-1} (3.321×10^{-2})	4.730×10^{-1} (1.595×10^{-1})	3.802×10^{-1} (1.124×10^{-1})	5.702×10^{-1} (4.983×10^{-2})	5.891×10^{-1} (5.189×10^{-2})	5.710×10^{-1} (5.502×10^{-2})	5.431×10^{-1} (2.361×10^{-2})
DAS-CMOP3_10	6.179×10^{-1} (2.778×10^{-1})	6.188×10^{-1} (2.519×10^{-1})	4.351×10^{-1} (1.452×10^{-2})	4.358×10^{-1} (2.089×10^{-2})	4.361×10^{-1} (6.076×10^{-3})	6.571×10^{-1} (2.194×10^{-1})	4.368×10^{-1} (3.464×10^{-2})	4.509×10^{-1} (3.314×10^{-2})	5.792×10^{-1} (2.037×10^{-1})	5.347×10^{-1} (2.851×10^{-1})	**8.020×10^{-1}** **(1.968×10^{-2})**	4.113×10^{-1} (2.367×10^{-2})
DAS-CMOP3_11	3.009×10^{-1} (1.534×10^{-2})	3.009×10^{-1} (1.534×10^{-2})	3.009×10^{-1} (1.534×10^{-2})	3.009×10^{-1} (1.534×10^{-2})	3.009×10^{-1} (1.534×10^{-2})	3.009×10^{-1} (1.534×10^{-2})	3.001×10^{-1} (1.393×10^{-2})	3.020×10^{-1} (1.496×10^{-2})	3.009×10^{-1} (1.534×10^{-2})	3.009×10^{-1} (1.534×10^{-2})	**3.046×10^{-1}** **(1.328×10^{-2})**	2.558×10^{-1} (2.874×10^{-2})
DAS-CMOP3_12	0.000 (0.000)	0.000 (0.000)	5.125×10^{-1} (3.240×10^{-2})	**5.131×10^{-1}** **(8.189×10^{-4})**	5.130×10^{-1} (5.335×10^{-4})	4.894×10^{-1} (5.862×10^{-2})	5.116×10^{-1} (4.614×10^{-2})	5.123×10^{-1} (1.342×10^{-3})	5.130×10^{-1} (7.730×10^{-4})	0.000 (0.000)	0.000 (0.000)	4.398×10^{-1} (9.680×10^{-2})

Table 4.5: (Continued)

	CDP	aE	E_1	cE_1	cE_2	ipE_1	iE_1	iE_2	pE_1	pE_2	SR_1	SR_2
DAS-CMOP4_9	5.241×10^{-1} (1.025×10^{-4})	5.241×10^{-1} (1.017×10^{-4})	**5.241×10^{-1}** **(8.775×10^{-5})**	5.241×10^{-1} (1.413×10^{-4})	5.241×10^{-1} (7.521×10^{-5})	5.241×10^{-1} (1.025×10^{-4})	5.241×10^{-1} (1.132×10^{-4})	5.240×10^{-1} (1.649×10^{-4})	5.241×10^{-1} (9.885×10^{-5})	5.241×10^{-1} (1.593×10^{-4})	5.240×10^{-1} (1.249×10^{-4})	5.240×10^{-1} (1.082×10^{-4})
DAS-CMOP4_10	5.506×10^{-1} (1.521×10^{-4})	5.506×10^{-1} (2.722×10^{-4})	5.499×10^{-1} (4.074×10^{-4})	5.503×10^{-1} (4.732×10^{-4})	5.494×10^{-1} (1.645×10^{-3})	5.505×10^{-1} (2.623×10^{-4})	5.500×10^{-1} (3.816×10^{-4})	5.495×10^{-1} (8.347×10^{-4})	5.504×10^{-1} (3.723×10^{-4})	5.505×10^{-1} (4.684×10^{-4})	**5.508×10^{-1}** **(3.158×10^{-4})**	5.446×10^{-1} (7.081×10^{-4})
DAS-CMOP4_11	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.271×10^{-1} (3.023×10^{-1})	6.717×10^{-1} (1.267×10^{-1})	**9.981×10^{-1}** **(7.569×10^{-5})**
DAS-CMOP4_12	3.243×10^{-1} (1.653×10^{-4})	3.243×10^{-1} (1.241×10^{-4})	3.245×10^{-1} (8.727×10^{-2})	3.243×10^{-1} (1.626×10^{-4})	3.244×10^{-1} (8.709×10^{-2})	3.243×10^{-1} (2.587×10^{-4})	**4.113×10^{-1}** **(8.167×10^{-4})**	4.087×10^{-1} (4.315×10^{-3})	3.243×10^{-1} (1.669×10^{-4})	3.243×10^{-1} (1.868×10^{-4})	3.243×10^{-1} (1.360×10^{-4})	4.026×10^{-1} (2.285×10^{-3})
DAS-CMOP5_9	8.258×10^{-1} (1.556×10^{-4})	**8.258×10^{-1}** **(1.468×10^{-4})**	8.258×10^{-1} (1.692×10^{-4})	8.258×10^{-1} (1.196×10^{-4})	8.258×10^{-1} (1.684×10^{-4})	8.258×10^{-1} (1.556×10^{-4})	8.258×10^{-1} (1.909×10^{-4})	8.258×10^{-1} (1.184×10^{-4})	8.258×10^{-1} (8.928×10^{-5})	8.258×10^{-1} (1.506×10^{-4})	8.258×10^{-1} (1.228×10^{-4})	8.258×10^{-1} (1.143×10^{-4})
DAS-CMOP5_10	8.794×10^{-1} (1.819×10^{-4})	8.794×10^{-1} (2.491×10^{-4})	8.788×10^{-1} (3.528×10^{-4})	8.791×10^{-1} (2.556×10^{-4})	8.782×10^{-1} (6.913×10^{-4})	8.792×10^{-1} (3.418×10^{-4})	8.788×10^{-1} (4.928×10^{-4})	8.784×10^{-1} (6.530×10^{-4})	8.791×10^{-1} (2.815×10^{-4})	8.793×10^{-1} (2.574×10^{-4})	**8.796×10^{-1}** **(2.829×10^{-4})**	8.731×10^{-1} (5.566×10^{-4})
DAS-CMOP5_11	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	6.425×10^{-1} (1.916×10^{-2})	1.121 (3.065×10^{-2})	**1.122** **(4.366×10^{-5})**
DAS-CMOP5_12	7.562×10^{-1} (7.565×10^{-1})	7.561×10^{-1} (7.564×10^{-1})	7.556×10^{-1} (7.730×10^{-4})	7.550×10^{-1} (7.563×10^{-1})	7.553×10^{-1} (1.134×10^{-3})	**7.563×10^{-1}** **(9.142×10^{-4})**	7.556×10^{-1} (8.017×10^{-4})	7.560×10^{-1} (2.213×10^{-3})	7.559×10^{-1} (7.563×10^{-1})	7.561×10^{-1} (1.908×10^{-3})	7.560×10^{-1} (7.565×10^{-1})	7.477×10^{-1} (1.314×10^{-3})
DAS-CMOP6_9	7.081×10^{-1} (8.724×10^{-5})	7.080×10^{-1} (8.320×10^{-5})	7.080×10^{-1} (9.158×10^{-5})	7.080×10^{-1} (9.029×10^{-5})	7.081×10^{-1} (1.133×10^{-4})	7.081×10^{-1} (8.724×10^{-5})	**7.081×10^{-1}** **(8.011×10^{-5})**	7.080×10^{-1} (1.274×10^{-4})	7.080×10^{-1} (1.794×10^{-4})	7.081×10^{-1} (8.080×10^{-5})	7.080×10^{-1} (1.486×10^{-4})	7.075×10^{-1} (4.322×10^{-4})
DAS-CMOP6_10	7.727×10^{-1} (1.350×10^{-2})	7.784×10^{-1} (1.213×10^{-2})	7.843×10^{-1} (1.195×10^{-2})	7.723×10^{-1} (1.508×10^{-2})	**7.851×10^{-1}** **(1.364×10^{-3})**	7.838×10^{-1} (1.227×10^{-2})	7.841×10^{-1} (1.184×10^{-2})	7.842×10^{-1} (1.199×10^{-2})	7.731×10^{-1} (1.239×10^{-2})	7.728×10^{-1} (1.229×10^{-2})	7.848×10^{-1} (1.221×10^{-2})	7.799×10^{-1} (4.440×10^{-3})
DAS-CMOP6_11	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	5.839×10^{-1} (2.149×10^{-2})	7.434×10^{-1} (3.081×10^{-1})	**1.075** **(1.384×10^{-4})**
DAS-CMOP6_12	0.000 (1.341×10^{-1})	0.000 (4.389×10^{-1})	8.046×10^{-1} (2.396×10^{-2})	6.844×10^{-1} (7.939×10^{-1})	7.702×10^{-1} (1.108×10^{-1})	7.694×10^{-1} (1.098×10^{-1})	**8.184×10^{-1}** **(6.516×10^{-4})**	8.126×10^{-1} (1.462×10^{-2})	5.861×10^{-2} (7.934×10^{-1})	5.446×10^{-2} (6.842×10^{-1})	0.000 (6.843×10^{-1})	7.990×10^{-1} (4.390×10^{-3})
DAS-CMOP7_9	1.122 (6.439×10^{-4})	1.122 (9.331×10^{-4})	**1.122** **(7.975×10^{-4})**	1.122 (8.109×10^{-4})	1.122 (6.892×10^{-4})	1.122 (6.439×10^{-4})	1.122 (1.137×10^{-3})	1.122 (1.051×10^{-3})	1.122 (9.146×10^{-4})	1.122 (4.627×10^{-4})	1.121 (1.021×10^{-3})	1.115 (8.802×10^{-4})
DAS-CMOP7_10	1.165 (1.139×10^{-3})	1.165 (1.372×10^{-3})	1.161 (2.176×10^{-3})	1.163 (2.377×10^{-3})	1.155 (2.112×10^{-3})	1.164 (1.207×10^{-3})	1.161 (1.245×10^{-3})	1.159 (1.297×10^{-3})	1.163 (1.258×10^{-3})	1.165 (2.010×10^{-3})	**1.166** **(1.128×10^{-3})**	1.145 (8.343×10^{-4})
DAS-CMOP7_11	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	9.115×10^{-1} (1.879×10^{-3})	**9.123×10^{-1}** **(2.368×10^{-3})**	9.109×10^{-1} (2.270×10^{-3})
DAS-CMOP7_12	1.130 (1.416×10^{-3})	**1.130** **(1.483×10^{-3})**	1.125 (2.188×10^{-3})	1.128 (3.118×10^{-3})	1.120 (2.439×10^{-3})	1.129 (2.193×10^{-3})	1.125 (2.481×10^{-3})	1.122 (2.728×10^{-3})	1.128 (2.654×10^{-3})	1.130 (1.834×10^{-3})	1.130 (1.843×10^{-3})	1.074 (5.643×10^{-3})
DAS-CMOP8_9	7.948×10^{-1} (1.101×10^{-3})	7.949×10^{-1} (6.728×10^{-4})	7.948×10^{-1} (9.074×10^{-4})	7.948×10^{-1} (1.043×10^{-3})	**7.950×10^{-1}** **(1.493×10^{-3})**	7.948×10^{-1} (1.101×10^{-3})	7.947×10^{-1} (1.137×10^{-3})	7.946×10^{-1} (8.440×10^{-4})	7.949×10^{-1} (6.322×10^{-4})	7.946×10^{-1} (6.691×10^{-4})	7.942×10^{-1} (1.265×10^{-3})	7.882×10^{-1} (9.336×10^{-4})
DAS-CMOP8_10	8.245×10^{-1} (2.690×10^{-3})	8.253×10^{-1} (2.962×10^{-3})	8.194×10^{-1} (1.921×10^{-3})	8.227×10^{-1} (2.578×10^{-3})	8.137×10^{-1} (1.239×10^{-3})	8.243×10^{-1} (1.807×10^{-3})	8.190×10^{-1} (3.546×10^{-3})	8.159×10^{-1} (1.614×10^{-3})	8.234×10^{-1} (3.995×10^{-3})	8.234×10^{-1} (2.458×10^{-3})	**8.271×10^{-1}** **(1.755×10^{-3})**	7.973×10^{-1} (1.198×10^{-3})
DAS-CMOP8_11	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (1.420×10^{-3})	9.715×10^{-1} (9.042×10^{-4})	**9.718×10^{-1}** **(4.065×10^{-4})**
DAS-CMOP8_12	**8.198×10^{-1}** **(1.943×10^{-3})**	8.195×10^{-1} (1.145×10^{-3})	8.149×10^{-1} (1.834×10^{-3})	8.183×10^{-1} (2.891×10^{-3})	8.085×10^{-1} (4.399×10^{-4})	8.191×10^{-1} (1.412×10^{-3})	8.144×10^{-1} (1.800×10^{-3})	8.114×10^{-1} (2.155×10^{-3})	8.168×10^{-1} (2.259×10^{-3})	8.194×10^{-1} (2.619×10^{-3})	8.196×10^{-1} (9.691×10^{-4})	7.664×10^{-1} (3.741×10^{-3})
DAS-CMOP9_9	5.527×10^{-1} (1.337×10^{-1})	6.195×10^{-1} (1.879×10^{-1})	5.944×10^{-1} (1.392×10^{-1})	5.518×10^{-1} (1.671×10^{-1})	6.103×10^{-1} (1.825×10^{-1})	5.527×10^{-1} (1.337×10^{-1})	6.071×10^{-1} (1.579×10^{-1})	5.923×10^{-1} (1.192×10^{-1})	5.963×10^{-1} (2.679×10^{-1})	5.449×10^{-1} (2.109×10^{-1})	5.861×10^{-1} (1.532×10^{-1})	**7.276×10^{-1}** **(1.354×10^{-1})**
DAS-CMOP9_10	6.141×10^{-1} (1.624×10^{-1})	6.311×10^{-1} (1.573×10^{-1})	7.216×10^{-1} (2.296×10^{-1})	7.693×10^{-1} (2.551×10^{-1})	8.229×10^{-1} (2.216×10^{-1})	6.825×10^{-1} (1.792×10^{-1})	9.097×10^{-1} (4.364×10^{-2})	**9.104×10^{-1}** **(1.394×10^{-1})**	6.499×10^{-1} (1.100×10^{-1})	6.470×10^{-1} (1.579×10^{-1})	9.093×10^{-1} (3.672×10^{-3})	6.951×10^{-1} (1.762×10^{-1})
DAS-CMOP9_11	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.353×10^{-1} (8.890×10^{-3})	6.182×10^{-1} (1.891×10^{-2})	**6.379×10^{-1}** **(5.391×10^{-2})**
DAS-CMOP9_12	4.881×10^{-1} (1.063×10^{-1})	5.108×10^{-1} (1.058×10^{-1})	6.366×10^{-1} (1.020×10^{-1})	5.984×10^{-1} (9.935×10^{-2})	5.808×10^{-1} (1.547×10^{-1})	5.364×10^{-1} (1.338×10^{-1})	6.325×10^{-1} (1.200×10^{-1})	7.071×10^{-1} (1.906×10^{-1})	5.551×10^{-1} (1.250×10^{-1})	5.320×10^{-1} (1.129×10^{-1})	**7.931×10^{-1}** **(2.392×10^{-1})**	6.213×10^{-1} (1.086×10^{-1})

Table 4.5: (Continued)

	CDP	aE	E_1	cE_1	cE_2	ipE_1	iE_1	iE_2	pE_1	pE_2	SR_1	SR_2
MW1	8.088×10^{-1} (6.164×10^{-5})	8.088×10^{-1} (5.334×10^{-5})	8.088×10^{-1} (4.541×10^{-5})	8.088×10^{-1} (6.430×10^{-5})	8.088×10^{-1} (3.597×10^{-5})	8.088×10^{-1} (4.467×10^{-5})	**8.088×10^{-1}** **(2.711×10^{-5})**	8.088×10^{-1} (4.228×10^{-5})	8.088×10^{-1} (9.178×10^{-3})	8.088×10^{-1} (6.059×10^{-5})	8.088×10^{-1} (4.794×10^{-5})	8.088×10^{-1} (5.229×10^{-5})
MW2	7.211×10^{-1} (2.939×10^{-2})	7.210×10^{-1} (1.341×10^{-2})	7.371×10^{-1} (4.377×10^{-5})	7.371×10^{-1} (1.606×10^{-2})	7.371×10^{-1} (1.606×10^{-2})	7.211×10^{-1} (2.626×10^{-2})	**7.371×10^{-1}** **(1.607×10^{-2})**	7.371×10^{-1} (1.605×10^{-2})	7.371×10^{-1} (1.615×10^{-2})	7.211×10^{-1} (2.620×10^{-2})	7.210×10^{-1} (3.048×10^{-2})	7.291×10^{-1} (1.607×10^{-2})
MW3	8.005×10^{-1} (4.325×10^{-4})	8.006×10^{-1} (6.022×10^{-4})	**8.007×10^{-1}** **(1.902×10^{-4})**	8.006×10^{-1} (2.600×10^{-4})	8.006×10^{-1} (3.572×10^{-4})	8.005×10^{-1} (3.325×10^{-4})	8.006×10^{-1} (4.493×10^{-4})	8.006×10^{-1} (2.715×10^{-4})	8.004×10^{-1} (4.536×10^{-4})	8.003×10^{-1} (5.100×10^{-4})	8.001×10^{-1} (3.626×10^{-4})	7.996×10^{-1} (3.837×10^{-4})
MW4	1.267 (2.476×10^{-4})	1.267 (1.578×10^{-4})	1.267 (1.263×10^{-4})	1.267 (2.323×10^{-4})	1.267 (1.925×10^{-4})	1.267 (2.017×10^{-4})	**1.267** **(1.876×10^{-4})**	1.267 (2.414×10^{-4})	1.267 (1.725×10^{-4})	1.267 (2.163×10^{-4})	1.267 (1.943×10^{-4})	1.266 (2.171×10^{-4})
MW5	7.241×10^{-1} (4.239×10^{-3})	7.269×10^{-1} (8.109×10^{-3})	7.271×10^{-1} (3.291×10^{-3})	**7.277×10^{-1}** **(5.682×10^{-3})**	7.241×10^{-1} (3.283×10^{-3})	7.269×10^{-1} (4.534×10^{-3})	7.256×10^{-1} (1.945×10^{-3})	7.265×10^{-1} (1.043×10^{-3})	7.243×10^{-1} (4.926×10^{-3})	7.262×10^{-1} (6.990×10^{-3})	7.259×10^{-1} (3.719×10^{-3})	7.263×10^{-1} (8.316×10^{-4})
MW6	4.410×10^{-1} (1.956×10^{-2})	4.410×10^{-1} (1.128×10^{-2})	**4.521×10^{-1}** **(1.114×10^{-2})**	4.521×10^{-1} (1.115×10^{-2})	4.410×10^{-1} (1.114×10^{-2})	4.410×10^{-1} (2.240×10^{-2})	4.521×10^{-1} (1.114×10^{-2})	4.465×10^{-1} (1.114×10^{-2})	4.521×10^{-1} (1.114×10^{-2})	4.410×10^{-1} (2.231×10^{-2})	4.410×10^{-1} (2.240×10^{-2})	4.521×10^{-1} (1.113×10^{-2})
MW7	5.567×10^{-1} (3.001×10^{-4})	5.569×10^{-1} (4.701×10^{-4})	**5.571×10^{-1}** **(1.363×10^{-3})**	5.565×10^{-1} (4.073×10^{-4})	5.566×10^{-1} (5.007×10^{-4})	5.564×10^{-1} (5.080×10^{-4})	5.562×10^{-1} (1.019×10^{-3})	5.564×10^{-1} (7.525×10^{-4})	5.567×10^{-1} (4.598×10^{-4})	5.565×10^{-1} (6.988×10^{-4})	5.545×10^{-1} (2.977×10^{-3})	5.555×10^{-1} (5.239×10^{-4})
MW8	1.040 (1.279×10^{-2})	1.036 (1.269×10^{-2})	1.043 (5.249×10^{-5})	1.043 (3.053×10^{-5})	1.043 (6.273×10^{-3})	1.043 (6.275×10^{-3})	1.043 (6.303×10^{-3})	1.043 (6.253×10^{-3})	**1.043** **(2.708×10^{-5})**	1.043 (6.271×10^{-3})	1.036 (6.330×10^{-3})	1.041 (8.258×10^{-5})
MW9	9.266×10^{-1} (3.397×10^{-3})	9.258×10^{-1} (6.275×10^{-3})	9.411×10^{-1} (2.854×10^{-3})	9.334×10^{-1} (3.508×10^{-3})	**9.431×10^{-1}** **(2.787×10^{-3})**	9.267×10^{-1} (3.498×10^{-3})	9.420×10^{-1} (4.224×10^{-3})	9.423×10^{-1} (4.063×10^{-3})	9.261×10^{-1} (3.670×10^{-3})	9.271×10^{-1} (4.919×10^{-3})	9.346×10^{-1} (3.450×10^{-3})	9.416×10^{-1} (2.899×10^{-3})
MW10	6.353×10^{-1} (3.969×10^{-2})	6.394×10^{-1} (3.998×10^{-2})	6.544×10^{-1} (3.719×10^{-2})	5.416×10^{-1} (6.896×10^{-2})	6.780×10^{-1} (1.613×10^{-2})	6.542×10^{-1} (3.631×10^{-2})	6.717×10^{-1} (1.592×10^{-2})	6.793×10^{-1} (1.682×10^{-2})	5.933×10^{-1} (3.082×10^{-1})	6.356×10^{-1} (3.690×10^{-2})	6.349×10^{-1} (5.104×10^{-2})	**6.871×10^{-1}** **(1.586×10^{-2})**
MW11	5.626×10^{-1} (9.920×10^{-2})	5.608×10^{-1} (2.710×10^{-2})	**7.439×10^{-1}** **(8.897×10^{-4})**	7.426×10^{-1} (4.730×10^{-3})	7.412×10^{-1} (1.518×10^{-2})	5.575×10^{-1} (4.173×10^{-2})	7.431×10^{-1} (3.110×10^{-3})	7.429×10^{-1} (1.019×10^{-3})	5.629×10^{-1} (1.260×10^{-2})	5.597×10^{-1} (7.123×10^{-2})	5.656×10^{-1} (1.608×10^{-3})	7.390×10^{-1} (1.222×10^{-3})
MW12	9.666×10^{-1} (8.096×10^{-4})	9.665×10^{-1} (5.765×10^{-4})	9.667×10^{-1} (3.992×10^{-4})	9.666×10^{-1} (6.223×10^{-4})	9.664×10^{-1} (4.940×10^{-4})	9.665×10^{-1} (8.679×10^{-4})	**9.670×10^{-1}** **(2.643×10^{-4})**	9.668×10^{-1} (3.980×10^{-4})	9.665×10^{-1} (6.263×10^{-4})	9.666×10^{-1} (3.468×10^{-4})	9.666×10^{-1} (2.571×10^{-4})	9.651×10^{-1} (1.055×10^{-3})
MW13	5.832×10^{-1} (3.234×10^{-2})	5.844×10^{-1} (3.010×10^{-2})	6.081×10^{-1} (2.475×10^{-2})	6.081×10^{-1} (2.112×10^{-2})	6.079×10^{-1} (2.461×10^{-2})	5.900×10^{-1} (2.217×10^{-2})	**6.199×10^{-1}** **(1.217×10^{-2})**	6.196×10^{-1} (1.254×10^{-2})	6.084×10^{-1} (1.268×10^{-2})	5.844×10^{-1} (2.221×10^{-2})	5.151×10^{-1} (8.895×10^{-2})	6.077×10^{-1} (2.139×10^{-2})
MW14	6.842×10^{-1} (2.813×10^{-3})	6.834×10^{-1} (3.080×10^{-3})	6.859×10^{-1} (1.642×10^{-3})	6.858×10^{-1} (1.736×10^{-3})	6.859×10^{-1} (1.338×10^{-3})	6.841×10^{-1} (2.839×10^{-3})	**6.860×10^{-1}** **(1.791×10^{-3})**	6.859×10^{-1} (1.414×10^{-3})	6.842×10^{-1} (1.910×10^{-3})	6.843×10^{-1} (2.428×10^{-3})	6.846×10^{-1} (2.385×10^{-3})	6.857×10^{-1} (1.579×10^{-3})
Windturbine	**9.060×10^{-1}** **(1.486×10^{-2})**	8.996×10^{-1} (1.385×10^{-2})	8.821×10^{-1} (1.826×10^{-2})	9.023×10^{-1} (1.457×10^{-2})	9.058×10^{-1} (1.639×10^{-2})	8.991×10^{-1} (1.747×10^{-2})	8.329×10^{-1} (3.610×10^{-2})	8.140×10^{-1} (4.378×10^{-2})	8.853×10^{-1} (2.906×10^{-2})	8.946×10^{-1} (2.429×10^{-2})	8.493×10^{-1} (3.006×10^{-2})	0.000 (4.289×10^{-2})

DAS-CMOP7_11, DAS-CMOP8_11, MW5, and MW6—no significant difference between the methods can be detected. Additionally, for 15 problems, there are three configurations or more that share the best performance. This leaves 41 problems, for which one or two configurations are best. The statistical analysis mostly confirms the previously stated trends, expect for MW problems where E_1 instead of iE_1 is more commonly among the bests.

To obtain a global view of the results, the methods are ranked from the best to the worst using these statistical outcomes. A dense ranking approach is used and ties are given the same rank. With the ranks, family-wise means are calculated. The resulting ranks are reported in Table 4.7. Mean ranks offer a chance to identify versatile methods that have good performance over several individual problems and the closer to one the better the expected performance.

Across all problems, the configuration cE_1 followed by cE_2 seem to be the best compromises. However, the mean overall ranks of all algorithms are quite close to each other, except for SR_2, which seems performing well only on DAS-CMOP with complexity level 11. SR_1 has overall the second worst rank and performs best only on DAS-CMOP with complexity level 10. The original E_1 performs best on MW. The *iE* method obtains the best ranks on C3DTLZ4 and DAS-CMOP with complexity level 12. The population-based methods by Stanovov et al. [167] only obtain good ranks on DAS-CMOP with complexity level 9. Their per-constraint method does not perform well for any problem family, although it shares its main concept with *cE*. In particular, its performance is relatively poor on MODAct. There, the newly proposed *cE* and its configuration cE_2 obtains the best rank. Finally, the promise of the parameterless *aE* method is not delivered and its performance is disappointing.

In order to visualize the practical differences, the final populations achieving the best and the median hypervolume for four configurations (cE_1, cE_2, iE_1 and pE_2) are compared against the best-known Pareto fronts for two difficult problems from MODAct: CS4 and CT4, Figure 4.2. Key observations are:

1. Both *cE* configurations have good convergence and spread for CT4 even for the median case, while cE_2 is clearly outperforming all others on CS4. Fronts obtained by cE_1 remain however of engineering quality.
2. The fronts obtained by iE_1 are of very limited use to engineers. These results suggest that the algorithm got trapped on all runs in a sub-optimal feasible region (CS4) or was not able to maintain sufficient diversity (CT4).
3. The best front obtained by ipE_1 could be of engineering value. Yet, the outcome seems not sufficiently reliable.

Some insights into the mechanisms at play in the different methods can be gained by looking at their respective ϵ values. Figure 4.3 shows the mean ϵ value for E_1, cE_2, iE_1, ipE_1, and pE_2 along generations when optimizing CS4 and CT4. While by definition, ϵ could increase

Table 4.6: Results from the statistical comparison procedure summarizing for each method and problem how many times the method performed better/equal/worse than the others. For problems where the null hypothesis of the Kruskal-Wallis test could not be rejected, all methods are considered equivalent. Best performing methods are highlighted in gray.

	CDP	aE	E_1	cE_1	cE_2	ipE_1	iE_1	iE_2	pE_1	pE_2	SR_1	SR_2
C3DTLZ4	3/5/3	3/5/3	9/1/1	3/5/3	9/1/1	3/5/3	1/1/9	10/1/0	2/5/4	3/5/3	3/0/8	0/0/11
CS1	1/9/1	2/9/0	1/10/0	2/9/0	2/8/1	1/9/1	2/9/0	2/9/0	6/5/0	1/9/1	1/4/6	0/0/11
CS2	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	1/10/0	2/9/0	2/9/0	2/9/0	1/10/0	1/2/8	0/0/11
CS3	4/5/2	4/5/2	0/3/8	4/5/2	10/1/0	4/5/2	0/3/8	0/3/8	10/1/0	4/5/2	4/5/2	0/3/8
CS4	2/7/2	3/6/2	1/5/5	10/0/1	11/0/0	2/7/2	2/7/2	3/6/2	1/1/9	3/6/2	2/7/2	0/0/11
CT1	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	0/0/11
CT2	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/9/1	1/10/0	2/9/0	1/10/0	0/0/11
CT3	5/6/0	5/6/0	1/3/7	5/6/0	5/6/0	4/7/0	1/3/7	1/2/8	2/4/5	5/6/0	4/7/0	0/0/11
CT4	5/4/2	5/4/2	1/2/8	10/0/1	11/0/0	5/4/2	1/2/8	1/2/8	4/0/7	5/4/2	5/4/2	0/0/11
CTS1	6/5/0	7/4/0	1/4/6	7/4/0	7/4/0	7/4/0	1/2/8	1/2/8	3/2/6	6/1/4	3/2/6	0/0/11
CTS2	6/3/2	6/5/0	2/3/6	8/3/0	8/3/0	6/5/0	1/1/9	1/2/8	3/2/6	6/3/2	3/2/6	0/0/11
CTS3	1/10/0	1/10/0	1/8/2	7/4/0	7/4/0	1/10/0	1/8/2	1/8/2	1/8/2	1/8/2	1/8/2	0/0/11
CTS4	1/5/5	1/5/5	7/3/1	7/3/1	11/0/0	1/4/6	7/3/1	7/3/1	3/3/5	1/4/6	1/5/5	0/0/11
DAS-CMOP1_9	0/11/0	1/10/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	1/10/0	1/10/0	1/10/0	0/7/4
DAS-CMOP1_10	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/10/1	0/11/0	0/11/0	2/9/0	0/10/1
DAS-CMOP1_11	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
DAS-CMOP1_12	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
DAS-CMOP2_9	1/10/0	1/10/0	1/10/0	0/11/0	0/11/0	1/10/0	0/11/0	0/11/0	1/10/0	0/11/0	1/10/0	0/5/6
DAS-CMOP2_10	0/10/1	0/9/2	0/10/1	2/9/0	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	10/1/0	0/9/2
DAS-CMOP2_11	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	0/10/1	11/0/0
DAS-CMOP2_12	0/10/1	0/9/2	0/11/0	0/11/0	0/11/0	0/10/1	1/10/0	6/5/0	0/11/0	0/10/1	0/10/1	0/10/1
DAS-CMOP3_9	5/6/0	3/8/0	2/6/3	2/9/0	0/2/9	5/6/0	0/5/6	0/2/9	4/7/0	5/6/0	3/8/0	2/5/4
DAS-CMOP3_10	6/4/1	6/4/1	1/4/6	1/4/6	1/4/6	6/4/1	1/4/6	1/5/5	6/4/1	5/5/1	11/0/0	0/0/11
DAS-CMOP3_11	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	0/0/11
DAS-CMOP3_12	0/3/8	0/3/8	5/6/0	5/6/0	7/4/0	4/5/2	5/5/1	6/5/0	5/6/0	0/3/8	0/3/8	4/1/6
DAS-CMOP4_9	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
DAS-CMOP4_10	6/5/0	6/5/0	1/3/7	2/6/3	1/9/1	4/7/0	1/4/6	1/5/5	3/5/3	4/7/0	7/4/0	0/0/11
DAS-CMOP4_11	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	10/0/1	11/0/0
DAS-CMOP4_12	0/8/3	0/8/3	0/11/0	0/8/3	0/10/1	0/8/3	10/1/0	8/2/1	0/8/3	0/8/3	0/8/3	7/2/2
DAS-CMOP5_9	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
DAS-CMOP5_10	7/4/0	7/4/0	1/4/6	3/5/3	1/5/5	5/5/1	1/5/5	1/3/7	2/6/3	5/5/1	9/2/0	0/0/11
DAS-CMOP5_11	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	10/0/1	11/0/0
DAS-CMOP5_12	0/11/0	0/11/0	0/10/1	0/11/0	1/10/0	1/10/0	1/10/0	1/10/0	0/11/0	0/10/1	0/11/0	2/5/4
DAS-CMOP6_9	1/10/0	1/10/0	0/10/1	0/10/1	1/10/0	1/10/0	1/10/0	0/10/1	0/10/1	1/10/0	0/10/1	5/0/6
DAS-CMOP6_10	0/10/1	0/10/1	0/11/0	0/10/1	6/5/0	0/11/0	0/11/0	0/11/0	0/10/1	0/10/1	0/11/0	0/10/1
DAS-CMOP6_11	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	0/9/2	10/0/1	11/0/0
DAS-CMOP6_12	0/6/5	0/5/6	6/3/2	0/7/4	1/7/3	5/3/3	11/0/0	9/1/1	0/7/4	0/6/5	0/6/5	8/1/2
DAS-CMOP7_9	1/10/0	1/10/0	2/9/0	2/9/0	2/9/0	1/10/0	1/10/0	1/10/0	2/9/0	2/9/0	1/5/5	0/0/11
DAS-CMOP7_10	7/3/1	7/3/1	3/1/7	5/3/3	1/0/10	5/5/1	3/1/7	2/0/9	5/3/3	5/5/1	11/0/0	0/0/11
DAS-CMOP7_11	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
DAS-CMOP7_12	8/3/0	8/3/0	3/1/7	5/4/2	1/1/9	5/4/2	3/1/7	1/1/9	5/3/3	5/6/0	6/5/0	0/0/11
DAS-CMOP8_9	1/10/0	2/9/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	2/9/0	1/10/0	1/8/2	0/0/11
DAS-CMOP8_10	5/5/1	6/4/1	3/1/7	5/4/2	1/0/10	5/5/1	3/1/7	2/0/9	5/5/1	5/5/1	11/0/0	0/0/11
DAS-CMOP8_11	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
DAS-CMOP8_12	6/5/0	6/5/0	3/1/7	5/6/0	1/0/10	5/6/0	3/1/7	2/0/9	5/2/4	6/5/0	6/5/0	0/0/11
DAS-CMOP9_9	0/10/1	0/11/0	0/11/0	0/10/1	0/10/1	0/10/1	0/11/0	0/10/1	0/11/0	0/10/1	0/10/1	7/4/0
DAS-CMOP9_10	0/8/3	0/8/3	0/10/1	0/10/1	0/10/1	0/8/3	6/5/0	6/5/0	0/8/3	0/8/3	9/2/0	0/8/3
DAS-CMOP9_11	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	0/1/10	0/11/0
DAS-CMOP9_12	0/6/5	0/6/5	4/7/0	0/9/2	0/9/2	0/6/5	4/7/0	7/4/0	0/9/2	0/6/5	7/4/0	4/7/0
MW1	0/11/0	0/11/0	0/11/0	0/11/0	1/10/0	0/11/0	2/9/0	2/9/0	0/9/2	0/11/0	0/8/3	0/11/0
MW2	0/10/1	0/6/5	3/8/0	4/7/0	1/10/0	0/11/0	1/10/0	1/10/0	0/11/0	0/9/2	0/9/2	0/11/0
MW3	2/9/0	1/10/0	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	1/10/0	1/10/0	1/3/7	0/0/11
MW4	1/10/0	1/10/0	2/9/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/10/0	1/9/1	0/0/11
MW5	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
MW6	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0	0/11/0
MW7	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	2/9/0	0/1/10	0/1/10
MW8	0/11/0	0/11/0	4/7/0	5/6/0	0/7/4	1/10/0	0/9/2	0/8/3	5/6/0	2/9/0	0/6/5	0/8/3
MW9	1/4/6	1/4/6	7/4/0	5/1/5	7/4/0	1/4/6	7/4/0	7/4/0	1/4/6	1/4/6	0/1/10	7/4/0
MW10	1/6/4	1/6/4	1/8/2	0/1/10	6/5/0	1/9/1	6/5/0	7/4/0	0/7/4	1/6/4	1/6/4	8/3/0
MW11	0/4/7	0/4/7	11/0/0	6/4/1	6/4/1	0/4/7	7/3/1	7/3/1	0/5/6	0/4/7	4/1/6	6/2/3
MW12	1/9/1	0/9/2	1/10/0	1/9/1	1/9/1	1/9/1	10/1/0	1/9/1	1/9/1	1/9/1	1/9/1	1/0/10
MW13	0/4/7	1/5/5	5/6/0	3/8/0	3/8/0	1/6/4	5/6/0	5/6/0	5/6/0	1/3/7	0/1/10	4/7/0
MW14	0/11/0	0/5/6	3/8/0	1/10/0	3/8/0	0/9/2	1/10/0	2/9/0	0/11/0	0/11/0	0/8/3	1/10/0
Windturbine	6/5/0	5/6/0	4/3/4	5/6/0	6/5/0	4/7/0	1/2/8	1/1/9	4/5/2	4/7/0	2/1/8	0/0/11

Table 4.7: Mean rank for each CHS per problem family. Rank is dense and ties are given the same rank (smaller is better). Best ranked CHSs are marked in bold.

	C3DTLZ4	DAS-CMOP (9)	DAS-CMOP (10)	DAS-CMOP (11)	DAS-CMOP (12)	MODAct	MW	Windturbine	Overall
CDP	3.0	1.67	2.56	1.78	3.89	2.50	3.50	**1.0**	2.69
aE	3.0	1.56	2.56	1.78	4.11	2.00	3.93	2.0	2.72
E_1	2.0	2.00	4.56	1.78	3.44	3.33	**1.64**	5.0	2.75
cE_1	3.0	2.11	3.56	1.78	3.56	1.50	2.36	2.0	**2.42**
cE_2	2.0	2.44	4.44	1.78	3.78	**1.25**	2.36	**1.0**	2.55
ipE_1	3.0	1.67	2.78	1.78	3.67	2.58	3.07	3.0	2.64
iE_1	6.0	2.33	4.22	1.78	**2.56**	3.33	1.93	7.0	2.78
iE_2	**1.0**	2.67	4.56	1.78	**2.56**	3.33	2.00	8.0	2.83
pE_1	5.0	**1.44**	3.33	1.78	3.78	2.75	2.86	4.0	2.73
pE_2	3.0	1.56	3.00	1.78	4.22	2.67	3.43	3.0	2.83
SR_1	4.0	2.11	**1.11**	1.67	3.67	3.17	4.43	6.0	2.92
SR_2	7.0	2.67	6.11	**1.22**	3.89	5.33	2.93	9.0	3.84
Number of problems	1	9	9	9	9	12	14	1	64

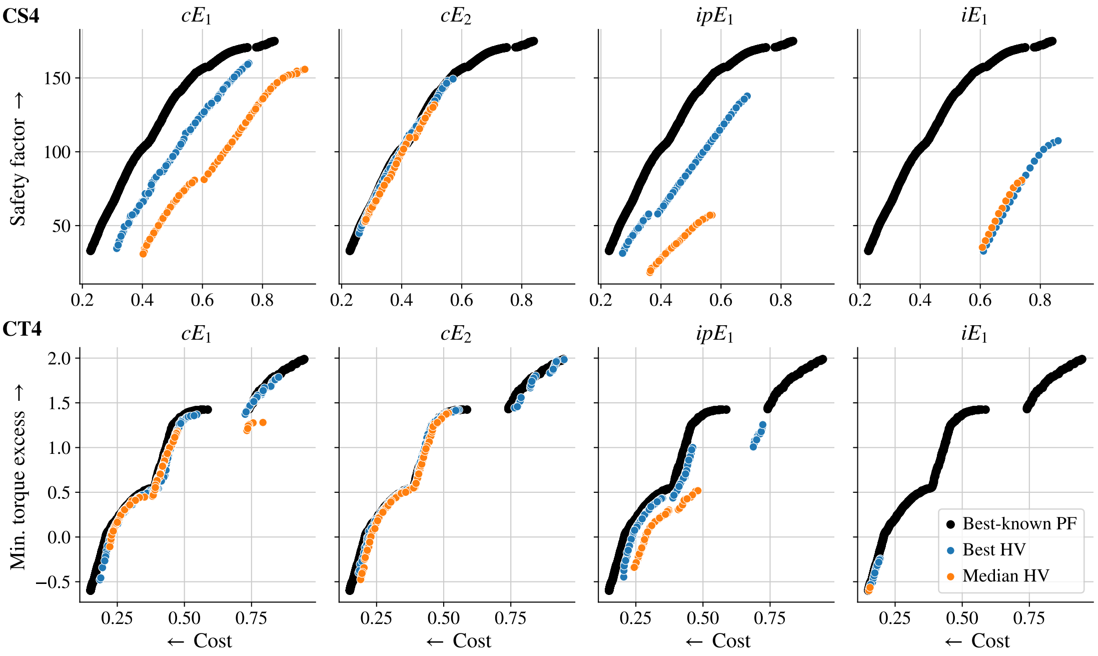


Figure 4.2: Comparison between the best and median final populations obtained by aE, cE_1, cE_2, and iE_1 and the best-known Pareto fronts for CS4 and CT4.

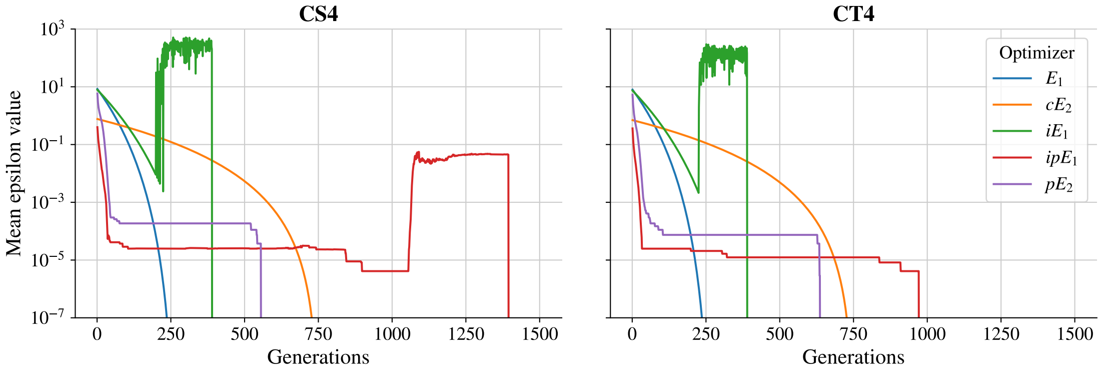


Figure 4.3: Evolution of the mean ϵ value in logarithmic scale across all optimization runs (and if applicable, across the multiple ϵ_j) for four configurations on CS4 and CT4.

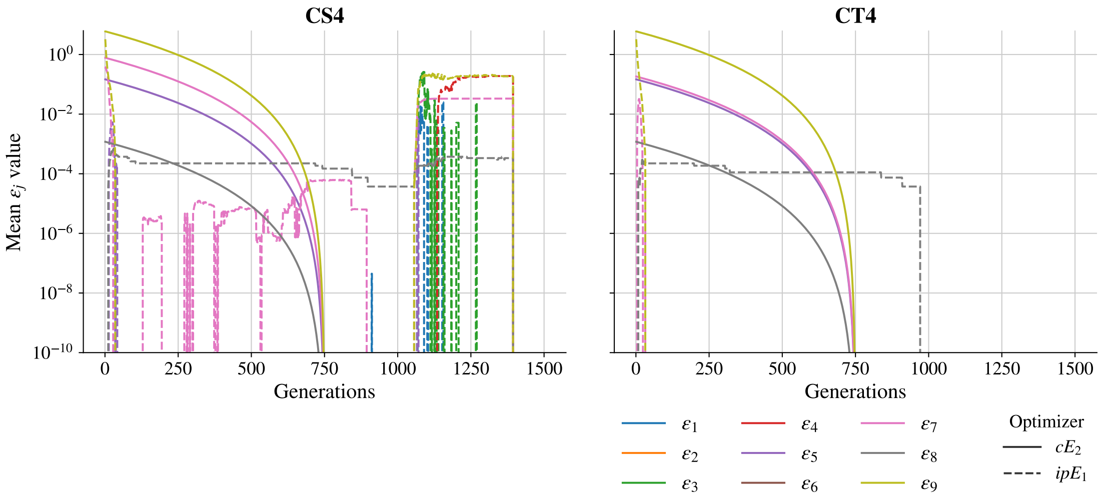


Figure 4.4: Evolution of the mean individual ϵ_j values in logarithmic scale across all optimization runs with cE_2 and ipE_1 of CS4 and CT4.

with iE, ipE, and pE, in practice, an increase is seen only for iE_1 on both problems and for ipE_1 for CS4. The update scheme of iE seems very sensitive, jumping from small to very high tolerances. Given the fronts shown in Figure 4.2, it seems that the method designed to allow the search process to get out of traps might not have worked as expected. The mean ϵ values of cE_1 while somewhat lower than expected from its γ follows the expected trend.

cE_2 and ipE_1 maintain individual ϵ_j values. Figure 4.4 shows the individual ϵ_j values for these two configurations for the same two problems. The interesting observation is that not all constraints are relaxed. In particular, cE_2 only relaxes four constraints of the nine for these problems. Figure 4.4 also highlights the radically different approach of the two methods. While cE_2 has a strictly decreasing tolerance, whose scale is appropriately scaled, ipE_1 has per-constraint tolerance levels, but there is no imposed decrease of its values. Some constraints that were not relaxed previously can have high tolerances at the end of the search.

4.5 Concluding remarks

4.5.1 Discussion

The executed large-scale investigation of seven existing CHSs and a novel per-constraint ϵ strategy cEpsilon on 64 CMOPs has revealed important performance differences between the strategies for the different problems. The analysis performed by means of ranking shows that the per-constraint relaxation methods cE_2, cE_1, and ipE_1 form the top 3 in terms of overall performance. Grouping the problems by family reveals, however, important differences. The original ϵ-constrained method, for example, performed best on MW problems. Remembering the differences in characteristics of these problems highlighted in Chapter 3, it is no surprise to see that some methods are more effective at solving specific problems. DAS-CMOP problems with complexity level 11, for example, have a high "convergence-hardness" [51], which means the constraints are blocking convergence, but the optimal solutions are not affected by the constraints. Hence, an optimizer disregarding constraints, like SR_2, is the ideal candidate. This is however, an unrealistic scenario for 'real-world' design applications. Considering the 13 mechanical design problem, the novel cEpsilon method obtained the best rank, suggesting that an individually controlled relaxation of each constraint offers significant benefits.

The analysis of the ϵ value on two MODAct problems revealed that each method does indeed have a different relaxation method. The methods where ϵ is allowed to increase, worked better on problems C3DTLZ4 and DAS-CMOP complexity levels 9 and 12, but not for MODAct. At least, the mechanism that can lead to an increase of ϵ in pE and ipE did not appear in CT4. cE_2 starts with a lower tolerance, but has a slower decrease. While this is expected from the selected parameters, all configurations have been suggested by the parameter tuning procedure. In that sense, all methods were given the same opportunity to select competitive parameters. Further, the analysis of the per-constraint ϵ_j revealed that only four constraints were relaxed by cE_2 and confirmed that the cEpsilon guarantees a constraint appropriate scale for ϵ_j and a strict decrease of all constraints.

4.5.2 Path forward

The proposed cEpsilon method has proven to be a robust method across several families of problems with best performance on mechanical design problems. While it offers sensible improvements over existing constraint relaxation methods, it does not solve their core issue: numerous parameters. The use of high power computing capabilities—with at times up to 500 cores running in parallel—made it possible to run the tuning procedure and performance evaluation campaign within a reasonable time, even for the slower MODAct problems. In practice, however, this is an important limitation, especially for computationally expensive problems. Original ideas are needed that are effective on realistic benchmark problems and that are applicable in practice. One idea could be to more tightly integrate the constraint handling within multiple components of algorithms, such as adaptive search operators based on relaxation stage.

The reported results also reaffirm the need for better and more realistic benchmark problems. Indeed, the behavior of the evaluated CHSs differed between MODAct or the Windturbine design problem and the more traditional synthetic problems. As long as most proposed optimization techniques are tested only on the latter problems, their transfer into other disciplines and practice is limited.

Tout change et grandit en ces lieux.
Quel air pur !

Guillaume (Guillaume Tell, G. Rossini)

5

Educating Future Engineers

MACHINE INTELLIGENCE and its use in engineering might substantially impact the work of engineers in the coming years. The motto 'adapt or die' seems accurate in this context. Researchers stand at the front line of innovation and may be the most capable of foreseeing the changes ahead. When researchers are also educators, they can play an ideal and critical role in shaping the future of engineering education.

Regarding the skills needed by engineers, an analysis of the skills listed for engineering positions in occupational databases—like O*NET—reveals that professional skills represent nowadays already an important part of the required skills, and some future-looking studies predict that such skills will become increasingly important with automation and digitalization [57, 129]. Also being able to work with, or even cooperate with machines, is fundamental, and requires computational thinking skills [66].

Two studies have been designed and performed with the participation of students from several classes with the aim to address two sets of research questions. On the one hand, the first study investigates the learning of professional skills by students working on team-based engineering projects. On the other hand, the second study exposes students to the developed automated design tool for actuators from Chapter 2, with the objectives to observe their design process, and to evaluate their use of the tool.

After an introduction of the context of the studies and their participants, this chapter is split into two parts. Section 5.2 discusses the topic of professional skills, and starts with the background about *learning by doing* in projects and the challenges of assessing professional skills. Then, through the use of a questionnaire applied in a pre-post study design, the impact of two kinds of projects on the students' self-efficacy beliefs is assessed and discussed. Section 5.3 focuses on the effects of the use by students of the automated design tool. The study design and the online platform, through which students interact with the tool, are presented first. The analysis is then performed considering the students' reports and the traces they left in the platform.

5.1 Context of the studies

5.1.1 Involved courses and study design

A total of three classes taught by Prof. Schiffmann in the department of mechanical engineering of EPFL have been included in these two studies: “systèmes mécaniques” (mechanical systems, 2$^{\text{nd}}$ year Bachelor), “dynamique des systèmes mécaniques” (dynamics of mechanical systems, 3$^{\text{rd}}$ year Bachelor) and “applied mechanical design” (Master). Figure 5.1 shows an overview of the timelines associated with each course.

The two Bachelor (BA) courses are mandatory for all mechanical engineering students and cover engineering fundamentals. They include 14 weeks of classes (*ex cathedra* lectures and exercise sessions), a written exam and a structured in-course group project (groups of four to five) with a final report. The in-course project starts on week 10. The final report (one per group) needs to be submitted on week 14 and evaluates the group’s technical work. Teaching assistants are available every week to answer questions. The in-course project contributes to 20% of the students’ grade (one grade per report) and its workload is estimated to 1 ECTS (about 28 hours) for each student.

The Master (MA) course is an optional capstone-like design project course where students work on a product design task (open-ended problem) in groups of three to four for the whole duration of the course. The design task is common to all groups, making it similar to a design challenge. The students need to hand in 3 written reports (week 3, week 10 and week 17) and do an oral presentation in front of the class (week 14), contributing to 10%, 40%, 40% and 10% of their grade respectively. In addition to the product, the grading process reviews project planning, risk assessment, and design process, which can be tracked through the repeated assessment. In parallel to the project, the students receive six theoretical lectures: the design process (four lectures), project management, and creativity fostering. As a support, each group has a dedicated teaching assistant. They meet weekly for followups. The total workload is estimated to 4 ECTS (about 112 hours) for each student.

The three courses are used to evaluate the development of professional skills by students through their participation in in-course or capstone projects. The interest is to see if there are differences in the learning between mostly technical short in-course projects and longer running capstone projects focusing on process skills. While in both cases, the group-project environment forces the students to practice their professional skills, the MA course includes theoretical lectures and a grading process that also considers planning and risk assessment. The changes are investigated through a questionnaire in a pre-post study design and since both BA classes are built using the same approach, their results are grouped in this study.

The effects of automated design on the product selection, the design process and the approaches followed by students for the use of such tools are investigated in the frame of the MA course. For this purpose a single design task is given to the students, and they are asked

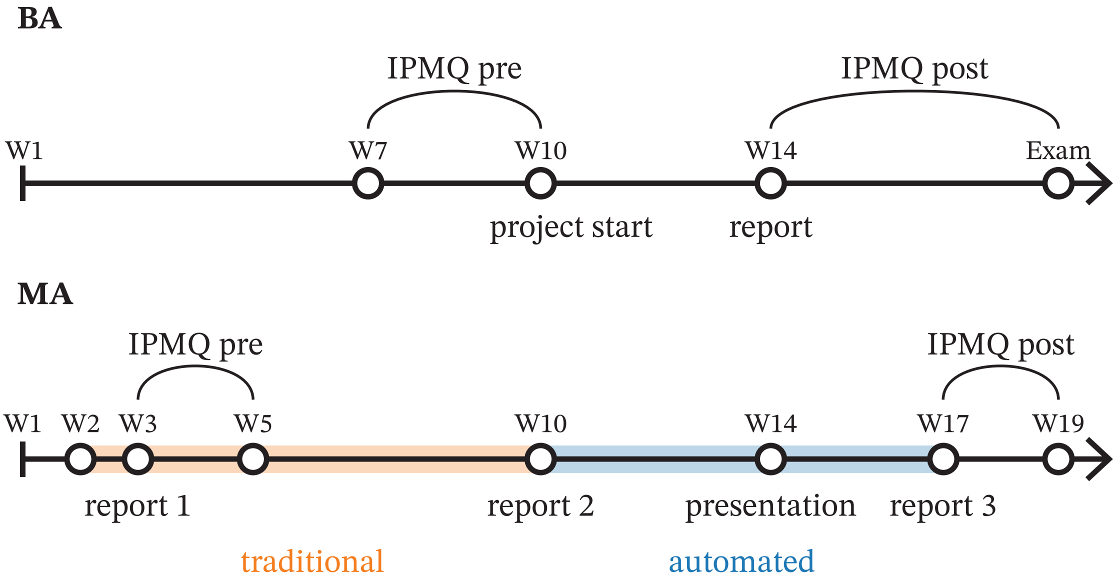


Figure 5.1: Overview of the timeline for the three courses highlighting the different phases and when the questionnaires were collected.

to provide two product propositions, once using a traditional approach and once using the automated design tool presented in Chapter 2. The timing of the two phases is shown in Figure 5.1. The traditional design phase starts on week 2 and ends on week 10 with the submission of the students' report containing their design. The automated design phase follows and ends on week 17 with the submission of the report containing the outcome of the automated phase. Other key milestones are report 1 occurring on week 3, which is used to clarify and formally validate the specifications, and the oral presentation on week 14 where the students present their designs to the rest of the class.

In order to contextualize the student data, their responses have been compared against those of experienced industry professional by having them fill the same questionnaire (professional skills) and by using an existing commercial product.

The specific details of the study design and the used instruments are given more in details in the corresponding sections.

5.1.2 Data collection protocol and population

Since the studies use data collected in the frame of teaching, students are required to give their informed consent for the use of their data for research. To ensure that the teaching is unaffected by the decision of students to participate or not in the study, the informed consents have been collected by Cécile Hardebolle from the teaching support center of EPFL and the consent information was only released after the grades had been validated. The corresponding data was anonymized by assigning each student a random ID. This protocol has been reviewed and approved by the EPFL Human Research Ethics Committee (HREC 046-2018).

The studies were performed in the fall semester 2018 (MA class) and in the spring semester 2019 (BA classes). From a total of 339 students, 205 gave their consent—174/306 BA and 33/33 MA students. All data produced by these students in the frame of teaching have been collected. These students did however not necessarily participate in all activities. The details of the collected data for each instrument are given in the corresponding sections.

5.2 Professional skills' development through projects

The work presented in this section stems from a collaboration between the Laboratory for Applied Mechanical Design (LAMD) and the Teaching Support Center (CAPE) of EPFL. The content reproduces, with minor stylistic changes and with the agreement of all authors, the following preprint:

C. Picard, C. Hardebolle, R. Tormey and J. Schiffmann. "Which Professional Skills do Students Learn in Engineering Team-Based Projects?". *Accepted for publication in the European Journal of Engineering Education*, April 2021.

Author contributions: CP, CH, RT, and JS designed research; CP and CH performed research; RT and CH wrote the related work; CP wrote the results and discussion.

In this section, we focus on the following research questions:

- Which professional skills do students learn from projects?
- How can we assess those skills and their evolution while taking into account scale issues of large-size engineering classes?
- Which characteristics of projects make students develop professional skills?

Using a previously validated instrument [99], we have assessed students' self-efficacy beliefs regarding their skills in five different areas: planning, risk assessment, ethical sensitivity, communication, and interprofessional competence. This instrument has allowed drawing comparisons across different levels of studies and careers, as well as assessing the progress students have made over the course of one specific project. Further, an analysis of students' answers to open-ended questions through a feedback survey has allowed investigating what non-technical skills students feel they have acquired from the projects and also which ones they have found to be the most challenging. Finally, comparing the format of the courses has led to the identification of some project characteristics, which might improve the development of professional skills for students.

In the following, a review of existing literature in the field is introduced, before present-ing the context and the methodology used. Finally, the results are presented and discussed before concluding.

5.2.1 Related work

In 2005, while reflecting on the importance of the professional skills included in the ABET "Engineering Criteria 2000", Shuman and his colleagues acknowledged the range of methods applicable to teaching professional skills [162]. The methods included decision-making exercises, project management or business simulations, project driven classes, case studies, as well as embedded modules. More recently, Winberg et al. [191] have attempted to classify these approaches in a systematic review of engineering employability studies. While reporting important variations in what the reviewed studies called "professional skills", the authors defend the idea that professional skills cannot be considered as generic but are instead linked to disciplinary practices. Therefore, they argue that engineering knowledge and professional skills should be better integrated.

Teaching professional skills through projects

As Prince and Felder [145] highlight in their review, projects used in teaching may vary significantly in scope and scale. From in-course to semester to BA or MA projects, the complexity of the problem, their duration and the size of the student group vary a lot. There are also variations in the ways in which projects are embedded in engineering curricula. This can be conceptualized as a continuum which ranges from projects being embedded in traditional engineering education programs in parallel to traditional courses, to projects being seen as central to a re-imagining of the whole engineering curricula. Some examples of the latter are the conceive-design-implement-operate (CDIO) approach or some variants of project/problem-based learning (PBL) [20, 30, 46, 95, 158]. In traditional program designs, projects frequently come at or near the end of a program through "capstone projects" or thesis [30], as a way to integrate learning and prepare students for the professional world. They are also increasingly being introduced earlier in the curriculum ("cornerstone" projects) as a way to scaffold students' understandings of real-life educational practice [65]. Although 'whole curriculum approaches' are influential within the literature and in the engineering education research community, they are perhaps less widely practiced than more traditional curricular approaches. Chen et al. [20], for example, found that two-thirds of the reviewed studies on project-based learning focused on projects within courses (rather than across courses or across whole curricula). Similarly, the large-scale cross-institutional survey of capstone design courses in the United States by Howe et al. [77] shows that most capstone projects are run with traditional lectures in parallel or before.

There is evidence that the inclusion of projects has a positive impact on some aspects of student learning. The presence of capstone-type projects has been found to generally improve employer satisfaction and employment ratios – possibly indicating an indirect effect on professional skills, but also student motivation and retention [76]. Studies within the PBL framework also point to an impact on some types of learning: Gijbels et al. [63] and Newman [127], for example, found that while problem-based learning had little impact on the learning of knowledge and facts, PBL did, however, have a notable and positive impact on understanding of principles and on the ability to apply what is learned.

Hattie's (2009) review of meta-analyses of learning provides a useful benchmark for evaluating such impacts: he argues that effect sizes (i.e., Cohen's d statistic) of less than 0.15 should be regarded as indicating effectively a lack of teaching, effects between 0.15 and 0.4 reflect normal teaching effects, and effects greater than 0.4 should be regarded as being in the desired zone. He noted few effects were greater than $d = 0.7$. The effect of PBL on ability to apply knowledge has been identified as $d = 0.4$, and for ability to use principles as $d = 0.75$ [63].

This highlights that one of the challenges with project-based learning is the validity of assessment – that is, are we assessing the things that we actually want to teach using project-based learning? There are numerous lists of professional or transversal skills to be learned by engineers. Accreditation bodies, for example, typically identify skills including ability to scope, plan, and design solutions to complex and ill-defined problems, ability to communicate, to work in teams, and to apply ethical reflection in their work (see for example, [47]). In their review of assessment tools for professional skills, Cruz et al. [31] focused on a similar set of skills: communication, lifelong learning, innovation/creativity and teamwork. Given that engineers are faced with complex problems that can often not be solved by any one discipline, interdisciplinary and interprofessional work has also been increasingly identified as an important professional competence [93, 100].

Conditions for learning professional skills during projects

One specific difficulty in projects comes from the tension between two different outcomes: the final product or solution on one hand, and student learning on the other hand. As Shuman et al. [162] highlight, professional skills have a strong "process" component, meaning that students need to learn about the processes behind communication, teamwork, project management, etc. Projects should therefore be an opportunity for students to focus their attention on such processes. This has not only been shown to be difficult by nature when "learning from doing" [8, 72] but can become even more challenging when projects are expected to lead to usable results, such as in industry-sponsored projects. Howe et al. [77] show that instructors are conscious about this issue since more than two thirds of respondents reported that they found the process to be equally or more important than the product. However, from the same survey, the *final* written report, the *final* oral presentation, and the *final product* play the biggest role when evaluating the students' work in projects. Whether such sources allow assessing how much students have developed process skills, is a key question that we will discuss in the next section. Another question is the type of support that can help students to focus their attention on the process during projects, thus fostering the development of professional skills.

In their review, Shuman et al. [162] argue that conditions for the effective development of process skills are that students should not be "thrown into team projects without support" but do not provide details about the instructional techniques behind such support. The same holds for the review by Winberg et al. [191], which underlines the importance of using adequate pedagogical approaches when integrating professional skills with engineering knowledge and skills. Unfortunately, the authors do not specifically review the associated instructional techniques. Overall, we were able to find much advice but very few studies looking at evidence concerning how professional skills development can be best supported during students projects. One such study, a large scale study by Cabrera et al. [13] has examined how teaching practices in team-based and hands-on design projects relate to students' gain in professional competencies. Using self-reported measures of both teaching practices and professional competencies as perceived by undergraduate engineering students, this multi-institutional study has shown that the practices used in class contributed more to the learning gains perceived by students than their background, demographics, or motivation. Interaction with and feedback from instructors as well as collaborative learning were the two instructional practices shown to predict gains in professional competencies, defined in this case as group skills, problem-solving skills and professional awareness. While very interesting by its scale, this study dates back almost twenty years and uses students as the only source of information, in a post-only survey. In a more recent but smaller qualitative study, Costa and her colleagues used focus groups to investigate the professional skills students considered to have developed during an interdisciplinary engineering project. The teaching techniques involved in the project were not the focus of the study but the authors do summarize a set of best teaching practices and conclude by emphasizing the critical importance of supervising and guiding students [27].

It was noted above that teamwork is an important engineering skill. There is evidence that the relatively homogeneous environment of engineering education makes student teams a potentially unpleasant environment for those who are not part of the majority group (e.g., [2]). The importance of guidance has been explored in studies focusing on teamwork skills more specifically [132, 144]. Because group work has the potential to lead to challenging experiences and outcomes for students (e.g., [83]), the explicit steps taken by instructors have been shown to be essential to assure learning. Reflection and self-assessment were shown to be effective techniques. Other studies focusing on engineering design projects, but not necessarily on professional skills, also underline the importance of guidance and supervision for student learning. Feedback, in particular, has been identified as a key factor for helping students learn by making them reflect on the process [8, 72].

The challenge of assessing professional skills

As introduced above, appropriate assessment methods are key to determine whether students do or do not develop professional skills in engineering design projects. However, several studies report that assessing professional skills is a challenge. Shuman et al. [162] state that efforts have been made to develop assessment tools for these skills but that "the literature remains sparse with respect to robust, effective measures for these outcomes". Some of the issues they identified are a lack of consensus on the definition of the skills, a difficulty to identify the moment when the skills should be assessed when learning is distributed over the curriculum, the very nature of the skills to assess and the cost in time and efforts when assessing those skills in large size classes. These difficulties continue to be acknowledged in more recent studies such as [13, 71, 144, 151], suggesting that this is a lingering concern.

Although there are a diversity of assessment methods [20, 77], the most widely used methods for assessing projects are final reports, presentations, and/or products. It is doubtful that these would allow valid assessment of students' professional skills. Project management skills such as planning illustrate this issue well: besides splitting the work into defined tasks, evaluating the time necessary to accomplish them, and organizing them along a timeline, appropriate planning requires a continuous adjustment, taking into account the evolution of the work as well as imponderables. A final project report alone is not likely to provide reliable information on the process through which students have made their plan evolve over time. More generally, the nature of professional skills makes the choice of the type of learning traces to collect, when to collect them, and the criteria and scales to evaluate them, challenging. In the context of projects, the adequate alignment between assessment methods, learning objectives and teaching methods is an additional difficulty [71, 191]. As a result, various other assessment formats have been explored in the literature, among which multi-source feedback (mixing peer feedback, self feedback, and instructor feedback [[121]]), portfolios – often in combination with rubrics [151, 191], and self-assessment questionnaires [71].

Cruz et al. [31] have recently mapped the range of assessment tools that can be used to assess some of the professional skills at issue in PBL: communication, lifelong learning, innovation/creativity and teamwork. They note that the kind of assessment used depends in part on the purpose of the assessment: where the goal is to assess students' learning, rubrics were the preferred assessment tool. However, where the goal is to assess the effectiveness of the course, questionnaires were commonly used. Such questionnaires typically rely on student self-reports of their own beliefs about their skills or their attitudes. Although Cruz et al. mention in passing their belief that self-report questionnaires are open to bias, this question has been subject to more detailed analysis in both psychological and statistical domains. From a psychometric point of view, it has been speculated that self-reports are biased when those with low skills in a domain are poor at assessing their own skill level in the domain (known as the Dunning-Kruger effect). More recent explorations of this apparent phenomenon suggest that this apparent bias is actually an artifact of the experiments designed to test it [130, 131]. They suggest that while self-reports are susceptible to measurement error (as with any measure), these measurement errors are not biased and are already taken into account in standard measures of reliability. From a psychological perspective, self-report measures are widely used in areas such as personality, metacognition, and self-efficacy. As Paulhus and Vazire [139] note, self-reports are popular in some domains in psychology not just because they are cheap and get a good response rate but also because they can provide access to information on the internal working of a person's psychological processes which would not otherwise be available, and because they assess how a person perceives something and this – rather than begin simply a source of bias – is actually a meaningful thing to measure in itself. This is particularly relevant in the case of self-efficacy beliefs, defined as a person's judgments of their own capabilities to undertake the actions needed to attain a designated goal. Students with higher self-efficacy beliefs have been found to choose more challenging tasks in the domain, to put in greater effort, to persist longer in the face of challenges, and to suffer reduced anxiety and stress. Self-efficacy beliefs also show a moderate to strong correlation with performance [203]. Overall then, there are good grounds for saying that self-report questionnaires may be a valuable assessment tool, if they are psychometrically assessed as valid and reliable (see also [13]). Where self-assessment tools are valid and reliable they can also have other benefits. Chen et al. [20] have highlighted that the challenges of assessing such learning in PBL are addressed both through more traditional assessment modes (such as quizzes and exams) as well as less traditional models (self-assessment, peer review, portfolios etc.). This, in turn raises problems for comparing learning from different PBL approaches – if all assessments are different (assessing different learning goals and with different psychometric characteristics) it is hard to draw meaningful conclusions about the impact on student learning of different ways of organizing PBL. The use of standard instruments means that it is possible to compare the impact of one type of pedagogy to another. This can provide valuable information to teachers who are seeking to enact pedagogies with demonstrated effectiveness.

It is worth noting that self-assessment questionnaires also have pedagogical advantages. In particular, responding to self-assessment questions can trigger students' self-reflection and draw their attention to what they learn, a key to "learning *from* doing" [8, 72]. Previous studies have shown that such tools could enhance learning and help students develop self-monitoring habits that are essential for the self-regulation of learning [161]. Panadero et al. [136] even propose that self-assessment tools originally designed for measurement could be considered as pedagogical interventions because of these effects.

Overall then, it is clear that professional skills are an important component of engineering education and these include the kinds of skills that are needed when working on complex and open-ended projects. Such skills include scoping, and organizing to design solutions to complex problems, integrating ethical considerations into one's work, communicating effectively in teams and working within interdisciplinary and interprofessional contexts. It is also clear that in engineering education, these skills are often targeted through projects—typically projects in courses. These seem to be organized in a wide variety of ways, even

Table 5.1: Summary of the different participants and the data collected.

Group	Study type	# Students	# Participants
Bachelor (BA)	IPMQ pre	306	168 (55%)
	IPMQ post	306	47 (15%)
	Survey	306	253 (83%)
Master (MA)	IPMQ pre	33	32 (97%)
	IPMQ post	33	29 (88%)
Professionals (Pro)	IPMQ single	–	40

if there is evidence that coaching, direction and feedback are all likely to be important in student learning. The assessment of different ways of organizing these projects in courses is made more complex by the use of a wide range of assessment instruments which may not actually address the competencies in question. This paper aims to contribute to this literature by exploring the differential impact of different approaches to organizing projects within a traditional engineering curriculum, using a psychometrically validated and reliable standardized assessment.

5.2.2 Methodology

Participants

Crossing the collected consents and questionnaires, 168 pre- and 47 post-questionnaires from BA students and 32 pre- and 29 post-questionnaires from MA students were collected. While the low number of respondents to the post questionnaire could imply biases in the data, the comparison of the answers to the pre questionnaires of the students that answered the post against the ones that did not, returned no statistical difference (two-sided independent t-test, $p = 0.27$). As such, it may be reasonable to hypothesize that this sub-group is representative of the wide class. In addition to the questionnaires, 253 anonymous student feedback forms were collected from the BA students.

Answers from 51 professionals were collected through anonymous survey links sent to engineering and technology companies collaborating with the university between Oct. 2019 and Jan. 2020. Based on the reported years of experience, professionals with less than 4 years of experience have been discarded (11 answers) to retain only the experienced ones.

The breakdown of participants is summarized in Table 5.1.

Instruments

The Interprofessional Project Management Questionnaire (IPMQ) was used as main quantitative instrument for this research. The IPMQ is a 24-item questionnaire available in two languages designed to assess one's self-efficacy beliefs in five domains: (A) planning, (B) risk assessment, (C) ethical sensitivity, (D) communication, and (E) interprofessional competence. This questionnaire was previously identified as having good factorial validity and reliability [99]. The questions and their factors are listed in appendix in Table B.1. Each question is evaluated with a 5-level Likert scale. The processing of the IPMQ implies the calculation of per-factor scores (average over the questions related to the factor) and a total score (average over all questions). After filling the questionnaire, the students received a feedback sheet summarizing their score with instructions on how to understand it and tips for further improving their skills. As such, the IPMQ can be considered as an educational intervention [161].

In addition to the IPMQ, data from anonymous student feedback have been collected from BA students. They were asked to report:

- "The 3 most important non-technical things I learned while carrying out the team project are" (open answer)
- "The 2 biggest challenges I encountered during the realization of the team project:" (open answer)
- "For the [first/second] of these challenges, I learned the skills to better manage a similar problem in the future" (two Likert questions associated to the previous question)

Research design and analysis

The students were asked to fill the IPMQ twice in a pre-post design. BA students could fill out the pre-questionnaire from week 7 until week 10 (start of project), and the post-questionnaire was open for answers starting after week 14 (end of project), see Figure 5.1. Additional questions have been added to the questionnaire to have students report their prior experience (pre) or gained experience (post). The students were asked to report the number of projects they had worked on, the number of project management courses followed, and the number of risk assessment courses followed. MA students were asked to fill the pre-questionnaire between weeks 3 and 5 (project started on week 1). The post-questionnaire was opened after the end of the project on week 17, see Figure 5.1. For the pre/post analysis, only the answers of students that responded to both questionnaires were considered.

Using the single questionnaire filled by professionals, a cross-sectional study comparing pre-scores of students against professionals was performed.

The statistical difference between two sets of answers is evaluated using independent (between population) or dependent (within population) t-tests. T-tests have been shown to be appropriate and robust for data from Likert scales even with moderate violation of normality [193]. When reporting the significance level of statistical tests, the APA star notation is

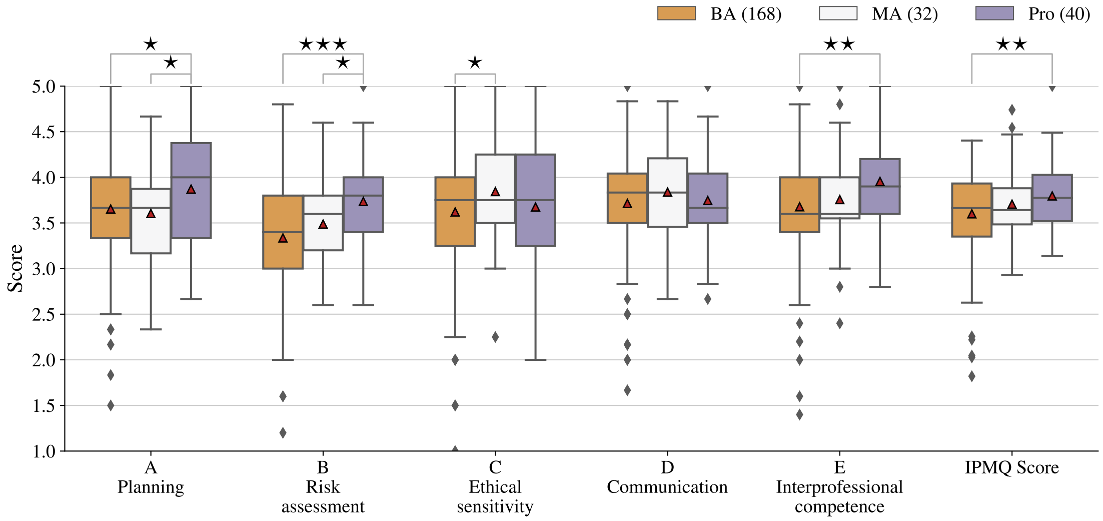


Figure 5.2: Cross-sectional comparison of pre-scores of students and professionals per factor and of the total score of the IPMQ. The 1.5 interquartile range (IQR) convention is used for the whiskers and the outlier identification. In addition, the mean value of the distributions is shown with a red triangle. Statistically significant differences between populations within factors are shown using the star notation, see Table 5.3.

Table 5.2: Mean and standard deviation (SD) of students' pre-scores and professionals' scores per factor of the IPMQ.

Factor	BA ($N = 168$)	MA ($N = 32$)	Pro ($N = 40$)
Planning	3.65 ($SD = 0.56$)	3.60 ($SD = 0.57$)	3.87 ($SD = 0.62$)
Risk Assessment	3.34 ($SD = 0.61$)	3.49 ($SD = 0.48$)	3.74 ($SD = 0.51$)
Ethical sensitivity	3.62 ($SD = 0.73$)	3.84 ($SD = 0.61$)	3.68 ($SD = 0.77$)
Communication	3.71 ($SD = 0.60$)	3.84 ($SD = 0.57$)	3.75 ($SD = 0.50$)
Interprofessional competence	3.68 ($SD = 0.59$)	3.76 ($SD = 0.54$)	3.96 ($SD = 0.49$)
Total IPMQ Score	3.60 ($SD = 0.46$)	3.71 ($SD = 0.41$)	3.80 ($SD = 0.39$)

used: * when $p < 0.05$, ** when $p < 0.01$ and *** when $p < 0.001$.

The open-ended survey questions have been analyzed using a coding approach. The codebook has been created through two iterations, ensuring a none overlapping coding and a common code for all questions.

5.2.3 Results

Cross-sectional view of professional skills

In a first step, the evolution of IPMQ scores across student levels to professionals is analyzed. For this analysis, only the pre-scores of the students are considered. Figure 5.2 is a box plot

Table 5.3: Results of the independent T-test statistical pairwise comparison between populations (BA, MA and Pro) within factors of the IPMQ with the alternative hypothesis that scores improve from BA to Pro (one-sided tail).

	BA vs MA			BA vs Pro			MA vs Pro		
Factor	T	DoF	p-value	T	DoF	p-value	T	DoF	p-value
Planning	0.442	43.08	0.6696	-2.033	55.11	* 0.0234	-1.892	68.55	* 0.0314
Risk Assessment	-1.567	52.17	0.0616	-4.282	68.33	*** 3.0e-5	-2.118	68.12	* 0.0189
Ethical sensitivity	-1.833	49.21	* 0.0364	-0.407	56.74	0.3427	1.038	70.00	0.8485
Communication	-1.124	44.97	0.1335	-0.343	67.61	0.3663	0.723	62.53	0.7638
Interprofessional competence	-0.743	46.25	0.2305	-3.083	68.73	** 0.0015	-1.612	63.37	0.0559
Total IPMQ Score	-1.310	47.23	0.0983	-2.762	67.84	** 0.0037	-0.950	64.88	0.1729

of the reported scores by population and factor and Table 5.2 reports the mean and standard deviation values.

Looking at the average scores per factor, it can be seen that there are differences and trends between students and professionals on the factors they are better at or worse. For students, risk assessment has the lowest score, while communication has the highest. For professionals, interprofessional competence has the highest score and ethical sensitivity the lowest.

More than the ranking of factors, it is the changes within factors from Bachelor students to professionals that are of interest. On the total IPMQ score, the mean value gradually increases, suggesting a progression from the first years of engineering education to professional practice. A similar trend is visible on planning, risk assessment, and interprofessional competence. Further, the lowest individual scores are within Bachelor students.

In order to confirm this trend, a statistical pairwise comparison between populations within factors is conducted and its results are reported in Table 5.3. This analysis confirms that there is a statistically significant increase between Bachelor students and professionals on planning ($p < 0.05$), risk assessment ($p < 0.001$), interprofessional competence ($p < 0.01$), and on the total IPMQ score ($p < 0.01$). There is also a significant increase between Master students and professionals on planning ($p < 0.05$) and risk assessment ($p < 0.05$). The difference between Bachelor and Master students is less clear. Statistically, they differ only on ethical sensitivity ($p < 0.05$).

Evaluation of an in-course project

Focusing on in-course projects first, their effect on the development of professional skills is assessed by running a pre-post comparison of IPMQ scores and by an analysis of course evaluation survey data.

Figure 5.3 shows the pre-post score distribution for the BA courses. The effective differ-

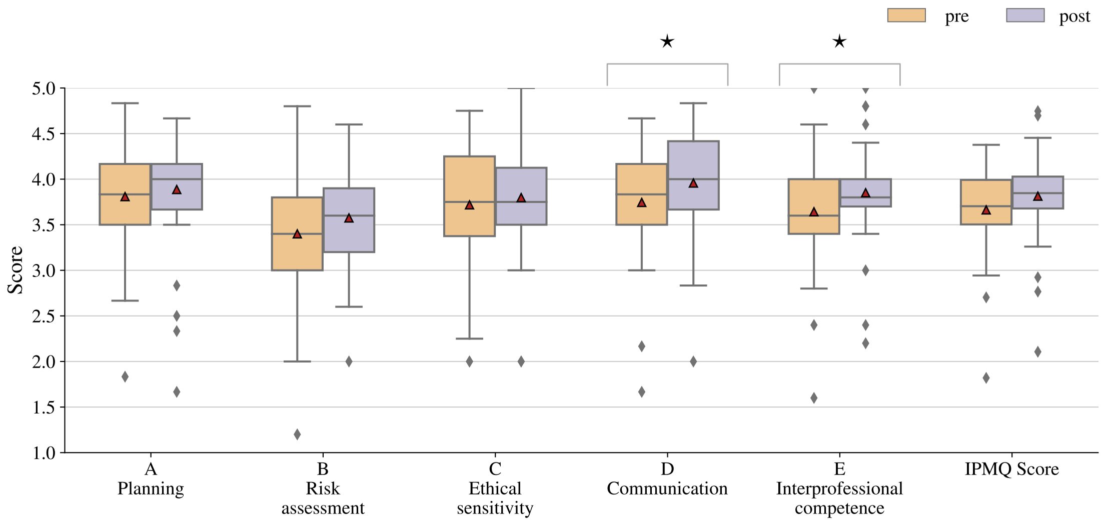


Figure 5.3: Box plot comparing pre/post scores of BA students per factor (mean value indicated by a red triangle). Only students that answered both questionnaires are considered ($N = 47$). Statistically significant differences within factors are shown using the star notation, see also Table 5.4.

Table 5.4: Mean and standard deviation, and results of the paired T-tests between pre/post scores per factor for BA students with the alternative hypothesis that post scores are better (one-sided tail). Only students that answered both questionnaires are considered ($N = 47$).

Factor	Mean pre-score	Mean post-score	T	p-value	Cohen's d
Planning	3.81 ($SD = 0.56$)	3.89 ($SD = 0.59$)	-0.697	0.2445	0.136
Risk Assessment	3.40 ($SD = 0.67$)	3.57 ($SD = 0.58$)	-1.538	0.0654	0.280
Ethical sensitivity	3.72 ($SD = 0.67$)	3.80 ($SD = 0.62$)	-0.714	0.2395	0.124
Communication	3.75 ($SD = 0.58$)	3.96 ($SD = 0.57$)	-1.897	* 0.0321	0.371
Interprofessional competence	3.64 ($SD = 0.59$)	3.85 ($SD = 0.51$)	-2.038	* 0.0237	0.379
Total IPMQ Score	3.66 ($SD = 0.46$)	3.81 ($SD = 0.45$)	-1.632	0.0547	0.333

ences within factors between pre and post has been statistically assessed and the analysis is summarized in Table 5.4, along with the mean and standard deviation for each factor.

The results show a positive trend on the total IPMQ score and a significant increase over the course of the project on interprofessional competence ($p < 0.05$) and communication ($p < 0.05$). However, with Cohen's d of 0.379 and 0.371 respectively, the effect sizes lie slightly below average for educational interventions. The other factors show no statistical difference.

While the primary objective of in-course projects is the development of technical skills, these projects are considered to be an important contribution to the development of professional skills. Yet, the results suggest none or limited impact of these projects on the learning of professional skills, even though students participated in several projects during the same

Table 5.5: Reported prior experience and training by BA students and evolution during the semester.

	Prior to the semester	During the semester
Number of projects	Median = 3; IQR = 3-4	Median = 2; IQR = 1-4
Courses on project management	Median = 0; IQR = 0-1	Median = 0; IQR = 0
Courses on risk assessment	Median = 0; IQR = 0-1	Median = 0; IQR = 0

time frame (Median 2, IQR = 1-4, see Table 5.5). During the same time, the students also reported that they had no additional courses addressing project management or risk assessment. As a consequence, the lack of explicit instructions and guidance is hypothesized as an important reason for the reported learning.

In complement to the IPMQ, the open-ended questions from the anonymous feedback surveys were processed. Since the survey is anonymous, the answers cannot be matched with the IPMQ, but they nevertheless offer additional insights into similar topics. The coded items reported by students in their top 3 of important non-technical things they learned are presented in Table 5.6, sorted by descending number of occurrences. Among the most commonly reported topics are communication (49), organization and coordination (46), time management (41) and task distribution (41). Although less frequent (29), team up challenges are also reported, inline with existing literature on difficulties of students in team projects [83]. With communication topics being reported first, the survey data are consistent with the significant increase and medium effect size found with the IPMQ on the communication factor.

The results of the second open question on challenges that students encountered and whether they feel they learned how to address them are reported in Table 5.7. There, the same aspects of planning are frequently reported as important challenges and have a relatively higher share of students disagreeing that they have learned how to address them.

Finally, inline with the learning goals of the project, mechanical engineering design methodology aspects are the most reported challenges, but have the highest share of students positive about their learning, confirming that the project achieves its primary goal.

Evaluation of a capstone project

Regarding the effect of capstone projects, the analysis is performed using a pre-post comparison of IPMQ scores. The results of the comparison are shown in Figure 5.4. The associated statistical tests including the mean and standard deviation are summarized in Table 5.8. Visually, there seems to be a clear increase in the reported scores for almost all factors. Statistically, planning ($p < 0.001$), risk assessment ($p < 0.05$), interprofessional competence ($p < 0.05$) and the total IPMQ score ($p < 0.01$) show a significant increase between the pre and post questionnaires.

Table 5.6: Analysis of students' answers to the open question "The 3 most important non-technical things I learned while carrying out the team project are:", N=136. The table presents only answers with more than 4 occurrences, grouped by topic with verbatim examples and ordered by decreasing frequency.

"The 3 most important non-technical things I learned while carrying out the team project are:"	# Occurrences
Communicate, share information	49
"Communication with other members of the group", "Good discussion in the group"	
Organize, coordinate, manage work	46
"Organize work", "Coordination"	
Manage time and workload	41
"Do a job with a deadline", "Plan", "Work regularly"	
Split and distribute tasks	41
"Share work", "Distribute tasks"	
Interpersonal attitude	35
"Trust others", "Be patient", "Listen to others", "Cope with others' motivation"	
Collaborate, cooperate, work in group	29
"Cooperate", "Work together", "Teamwork"	
Team up, get along	29
"Work with unknown people", "Get to know people", "Highlighting the qualities of each person is important"	
Mechanical engineering design methodology	10
"Check results", "Work specifications out"	
Anticipate and solve issues	5
"Learn to prevent issues", "Issues need to be addressed as soon as they occur"	

The gains on the planning and risk assessment factors and on the total IPMQ score show an effect size ($d = 0.538$, $d = 0.453$ and $d = 0.465$ respectively) above the 0.4 threshold, suggesting an above average effect of the educational interventions on professional skills.

A closer analysis reveals that this is actually notably affected by a single outlier who scored extremely low on the post test (e.g., risk assessment at 1.4). For example, if this student is excluded, the effect size for the risk analysis factor rises from a moderate effect ($d = 0.453$) to a very strong effect ($d = 0.718$) among educational interventions.

5.2.4 Discussion

Reverting to our research questions, we first look at our results in terms of the professional skills' development. In general, professional skills tend to be implicitly and not explicitly treated as part of traditional curricula [77]. For example, project planning skills are seen as a prerequisite for higher education (e.g., ABET), but are nevertheless only addressed in ad hoc classes. Indeed, both BA and MA students in our study showed a significantly lower score compared to professionals. So while, planning is required to deal with the project in class, the students don't seem to develop their skills through their studies in general. Further, neither students nor professionals seem to learn certain skills, such as ethical sensitivity, which has also been reported by Cech [17] or Tormey et al. [178]. This calls for a broadening

Table 5.7: Analysis of students' answers to the open question "The 2 biggest challenges I encountered during the realization of the team project:", combined with two Likert questions: "For the [first/second] of these challenges, I learned the skills to better manage a similar problem in the future". The number of students having answered all three questions is N=110. Answers are grouped by topic, presented with examples and ordered by decreasing number of occurrences on "Agree". Answers with 4 occurrences or fewer are not presented.

	"For the [first/second] of these challenges, I learned the skills to better manage a similar problem in the future"	
"The 2 biggest challenges I encountered during the realization of the team project:"	# Agree	# Disagree
Mechanical engineering design methodology "Understand the problem", "Gearbox dimensioning", "Calculate the forces"	68	6
Manage time and workload "Time management", "Meeting deadlines", "Work"	59	13
Split and distribute tasks "Allocation of work", "Split tasks"	55	13
Team up, get along "Working with unknown people", "Make sure everyone is involved in the project"	43	11
Organize, coordinate, manage work "Organize as a group", "Coordination", "Decision making"	24	5
Communicate, share information "Intra-group communication", "Information sharing"	21	3
Use software tools, Matlab "Code in Matlab", "Use Matlab", "Use of digital tools"	18	5
Interpersonal attitude "Trust", "Stay calm"	18	4
Availability of teaching assistants "Not enough assistants", "Have access to assistants"	6	1
Collaborate, cooperate, work in group "Work in group"	5	1

use of professional skills assessment tools as teacher reflective devices about the content of one's courses and as reflective devices for students to make the learning goals of projects explicit.

There has been a growing focus on the methods – and their quality – used to assess professional skills [31], which echoes our second question about assessment methods. In this study, we have evaluated the use of a standardized questionnaire: the IPMQ. The usefulness of the IPMQ as a measure of students' learning seems confirmed by the benchmark against experienced professionals. The latter indeed scored higher than students except in ethical sensitivity and communication. Their relative low score on ethical sensitivity may actually well reflect the reality of industry, where such questions are not necessarily the highest priority. Within courses, the IPMQ was able to capture different gains between in-course and capstone projects. Students increase a bit their self-efficacy beliefs on courses where profes-

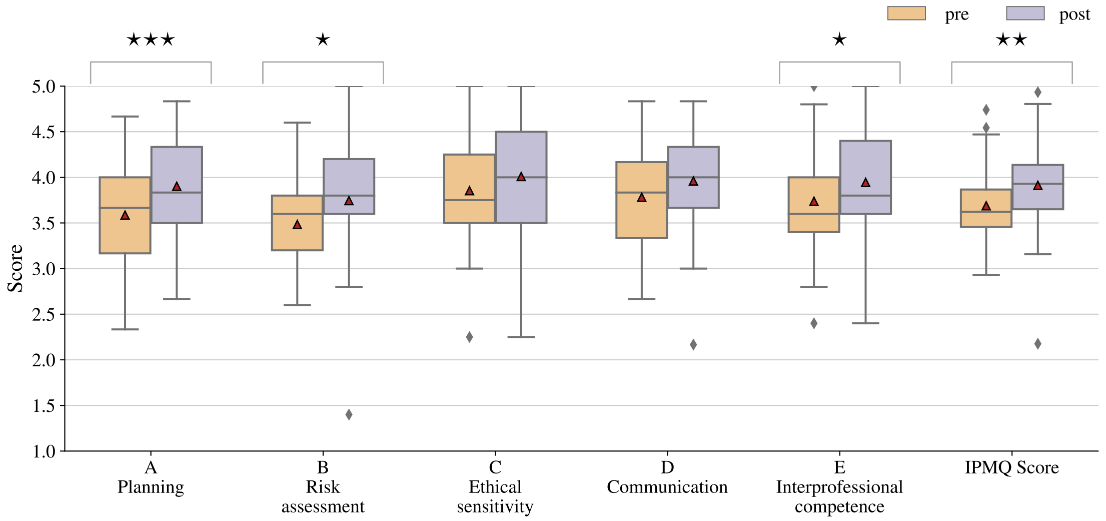


Figure 5.4: Box plot comparing pre/post scores of MA students per factor (mean as a red triangle). Only students that answered both questionnaires are considered ($N = 29$). Statistically significant differences within factors are shown using the star notation, see also Table 5.8.

sional skills can be learned, and more when they are explicitly taught.

Yet, the IPMQ may not be sensitive to all learning. In the BA courses, students reported, through their answers to a survey, learning coordination, time management and task splitting that correspond mostly to items from the planning factor of the IPMQ (define work plan Q2, breaking work into tasks Q3 and keep track of tasks Q5) on which there was no significant difference. This could be explained by the individual framing of IPMQ questions whereas students see those as group activities, or they feel their learning are not sufficient to state that “they are good at”.

While there have been recent publications discussing the characteristics of projects that make students develop professional skills more [27, 191], these are mostly prescriptive studies. With respect to that, our data specifically suggest that:

1. A broader learning is obtained from bigger and more complex projects than from smaller in-course projects.
2. A stronger learning is obtained when there is explicit teaching and regular formative and summative feedback on process skills.

There are, however, a few open questions and limitations to our study. The reported gains in self-efficacy beliefs of interprofessional competence are unexpected and cannot be related to the settings of the projects (all students were from the same department) nor was it part of the teaching. Since this could be due to a transfer by the students from interprofessional to intra-group interindividual concepts, it also highlights the challenges of self-assessment.

Table 5.8: Mean and standard deviation (SD), and results of the paired T-test statistical comparison of pre/post IPMQ scores per factor for MA students with the alternative hypothesis that post scores are better (one-sided tail). Only students that answered both questionnaires are considered ($N = 29$).

Factor	pre	post	T	p-value	Cohen's d
Planning	3.59 ($SD = 0.60$)	3.90 ($SD = 0.58$)	-3.622	*** 0.0006	0.538
Risk Assessment	3.48 ($SD = 0.50$)	3.75 ($SD = 0.66$)	-1.817	* 0.0400	0.453
Ethical sensitivity	3.85 ($SD = 0.63$)	4.01 ($SD = 0.70$)	-1.330	0.0971	0.234
Communication	3.78 ($SD = 0.56$)	3.96 ($SD = 0.58$)	-1.682	0.0518	0.312
Interprofessional competence	3.74 ($SD = 0.56$)	3.95 ($SD = 0.64$)	-2.030	* 0.0260	0.347
Total IPMQ Score	3.69 ($SD = 0.43$)	3.91 ($SD = 0.54$)	-2.583	** 0.0077	0.465

Indeed, while self-efficacy beliefs have been shown to correlate with performance [203], it is still unknown if this correlation stands for professional skills. The use of objective measures would limit this issue, but objective measures of performance in these domains that can standardize across disciplines and courses still need to be developed. Further, while Likert scale based self-assessment tools are simple to apply and score, motivating large classes to participate remains a challenge of its own and certainly even more if the questionnaires are not well aligned with the explicit learning objectives of courses.

5.2.5 Concluding remarks

While the results suggest that even in in-course projects student develop their communication and interprofessional skills, capstone projects seem to support a broader set of skills and a deeper development, in particular of risk assessment and planning skills. However, the results also highlight the remaining gap with respect to professionals, emphasizing the need for greater practice through the curriculum.

In terms of tool, the IPMQ proved to be capable of measuring differences in pre-post assessments. The collection, scoring, and report generation for student feedback can be highly automated, making it practical to assess in a standardized way the self-efficacy beliefs of many students. The challenge remains, however, the participation of the students in larger classes. The other shortcomings of the questionnaire are the lack of context specific aspects, highlighted by the students, including team-up and team organization issues.

Professional skills are gaining importance for future engineers and need to be included in modern curricula. However, simply replacing traditional classes by projects and implicitly expecting students to develop such skills without explicit teaching and adequate feedback is not supported by our data. Broad and strong learning can be achieved by explicitly including profession skills in the instructions and by providing multiple feedback loops. Such changes are possible today, without whole curriculum redesign, and should help better prepare the engineers of the 21^{st} century.

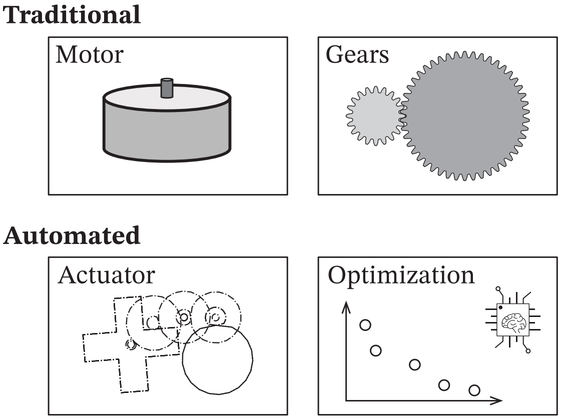


Figure 5.5: Overview of the models accessible to the students in each phase.

5.3 Use of machine intelligence by novice engineers

In this section, the consequences of the use of machine intelligence by novice engineers are investigated by observing its impact on the obtained designs and on the design process. The goals are to clarify the challenges faced by students, while also demonstrating the effectiveness of the automated design tool.

In the following, the study design is presented along with the Online MODelling platform (OMOD), through which students perform simulation requests. Then, the data analysis approach is detailed. Finally, the results are exposed by focusing first on the differences in terms of obtained actuators from the use of the automated tool, and second, on the processes applied by students.

5.3.1 Study design

In addition to more pronounced professional skills, modern engineers also need to be trained to interact and to take advantage of the growing machine intelligence tools. In order to gather evidence about these questions, the students of the MA design class have been given the task to design an electro-mechanical actuator for automotive HVAC flap control. The specifications given to the students are the same as an existing product and are summarized in Table 5.9.

The 33 students formed nine groups of three to four. They repeated the design task twice: once following a traditional approach and once using the developed automated design tool. At the end of each phase, each group submits one report—report 2 and report 3 respectively—which presents the design of their actuator, including detailed information and drawings. The students are given access to different models, see Figure 5.5, through a web app named OMOD, which also allows data collection. The platform is described in more details in Section 5.3.2.

Table 5.9: Relevant specifications for the actuator to be designed.

		Min	Nominal	Max	
	Output specifications				
C1	Operating temperature	-40	25	80	°C
C2	Voltage supply	9	14	16	V
C3	Operating torque (3 rpm)	0.4		0.6	N m
	Fatigue	10 000	cycles at −40 °C		
		50 000	cycles at 25 °C		
		20 000	cycles at 40 °C		
		10 000	cycles at 80 °C		
C4	Holding torque	1.21			N m
	Fatigue	1000	cycles at −40 °C		
		5000	cycles at 25 °C		
		2000	cycles at 40 °C		
		1000	cycles at 80 °C		
C5	Efficiency	60			%
C6	Motor drive			0.5	A
C7	Minimal resolution	6400			steps/rev
	Manufacturing / Assembly				
C8	Allowed processes	Plastic injection for gears (given list of materials)			
C9	Single component	Consecutive pinion and wheel are injected as a single part			
C10	Stepper motors	Chosen based on given list			
	Coil Fill factor	0.5		1.2	-
	Resistance scaling	0.5		1.2	-
C11	Assembly	Components are inserted sequentially in housing following the z (height) axis			
C12	Axis locations	Axis supports should be part of housing			
	System level				
C13	Housing dimensions	46 mm x 79 mm x 24 mm			
C14	Cost	To be minimized			

In the traditional design phase, students only had access to component models from the automated design tool and had to rely on their own tools and skills in particular for the system integration steps. This phase is considered “traditional”, since it corresponds to the most common state in many industries: numerical models are available for specific components and team work following a segregated approach. While, there was no rule that prohibited the use of optimization techniques, the setting made it difficult in the allotted time. In the automated design phase starting W10, students were given access to the actuator and the optimization module, and thus to the whole automated design tool.

In order to support students, a one hour lecture is given on week 2 to present OMOD, a two-hour lecture on week 9 to introduce the principles of automated design, and another two-hour lecture on week 10 to present the automated design tool in OMOD.

5.3.2 OMOD – User interface

The automated design tool and its models is made available through OMOD, which is a web interface built using the Python web framework: tornado[1]. It allows students to access complex software through a modern web browser requiring only a computer with internet connection. The web interface allows students to:

1. start new calculations—called 'evaluation'—of one of the four available models, see Figure 5.5;
2. visualize the results of previous evaluations;
3. download the data related to an evaluation for offline processing.

Each student from the course is granted individual access—managed through the school's central identity provider service—but each member of a group can view the results from evaluations of the other group members.

The general architecture of OMOD is illustrated in Figure 5.6. OMOD is coupled to a database to store the evaluation requests and their results, and acts as an intermediary between the students and the actual automated design code. Through a queue, it can dispatch calculations to "workers" and results become available when processing is completed. The results are shown as tabular data, interactive plots generated by bokeh[2] [11], and 3-D meshes using three.js[3] [32] depending on the type of evaluation.

To start motor or gear-pair evaluations, students fill out an online form asking for the parameters and operating conditions of the respective models. Due to the large number of parameters, actuator simulations and optimizations are started by submitting "job files". An example for each type has been provided to the students, and is shown in Appendix C. While this interface does not allow students to define arbitrary objectives and constraints, a generic function has been set up and can be customized through these job files. With over 30 variables to choose from to define the objectives (from two to three) and constraints, students can activate almost all options—available in 2018—of the integrated model. Some of these variables have even been added following specific requests by students. Students can

---

[1] https://www.tornadoweb.org

[2] https://bokeh.org

[3] https://threejs.org

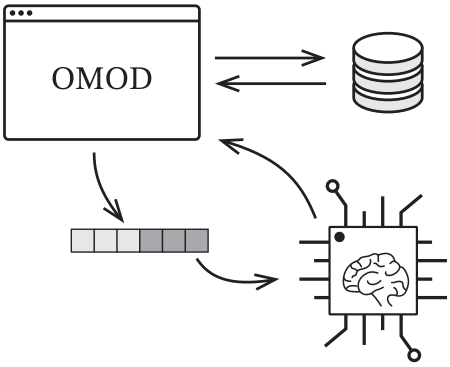


Figure 5.6: General architecture overview of the OMOD interface.

also configure the population size and the optimization budget—limited to 150 000 solution evaluations. Regarding the results, all available outputs from the models are returned—geometrical, kinematic, mechanical strength, system-level, and so on. For the optimizations, the results include graphs representing the convergence process (minimum and maximum CV values and hypervolume evolution), the Pareto front, a parallel coordinate plot and a table for the decision variables, and the meshes of all actuators. The data can be downloaded as JSON or CSV files for tabular results, interactive HTML for plots and Wavefront OBJ for meshes.

OMOD and its workers run on a single server featuring an Intel Xeon E5-1650 v4 (6 cores, 3.6 GHz) with 64 GB of RAM. There are two workers running in parallel to process all evaluation requests from the groups. A third worker was set up during the automated design phase to process all non-optimization requests to ensure small jobs would not get queued for hours. For the optimization, each worker can use up to 3 cores to evaluate the solutions in parallel. NSGA-II with adaptive search operators—the same as presented in Section 2.4—and the ϵ-constrained method ($\lambda = 0.5, c_p = 5, \gamma = 0.5$)[4] are used for the optimization.

5.3.3 Approach to the analysis of data

The analysis aims at investigating the changes in the design and also in the process followed by students during the two phases. The collected data for this purpose includes for each group: (i) three reports (R1, R2, and R3), (ii) the stored evaluations in OMOD, and (iii) the self-reported time sheets. Most analyses are performed at the group level where each group is identified by a group ID (GID).

The selected design for each group in each phase is extracted manually from their reports and converted into an actuator definition understandable by the automated design. The performance metrics of the actuators are evaluated using the same ‘actuator’ module that was available to students, but a unified set of operating points is used, see Table 5.10. This

---

[4] The better constraint handling strategy cEpsilon was not ready back then.

Table 5.10: Most critical operating points based on Table 5.9 used to evaluate the actuators and assess their performance.

Output	T_{desired}	Temp.	# cycles	V_s	$I_{\max}$	Condition
$0.314\,\text{rad}\,\text{s}^{-1}$	0.4 N m	−40 °C	10 000	9 V	0.5 A	running
$0.314\,\text{rad}\,\text{s}^{-1}$	0.4 N m	25 °C	50 000	9 V	0.5 A	running
$0.314\,\text{rad}\,\text{s}^{-1}$	0.4 N m	40 °C	20 000	9 V	0.5 A	running
$0.314\,\text{rad}\,\text{s}^{-1}$	0.4 N m	80 °C	10 000	9 V	0.5 A	running
0.0056 rad	1.21 N m	−40 °C	1000	9 V	0.5 A	holding
0.0056 rad	1.21 N m	25 °C	5000	9 V	0.5 A	holding
0.0056 rad	1.21 N m	40 °C	2000	9 V	0.5 A	holding
0.0056 rad	1.21 N m	80 °C	1000	9 V	0.5 A	holding

ensures that all solutions are evaluated on equal foot and that the actuators fulfill the minimum requirements from the specifications. For comparison purposes, the same process is applied to the existing industrial product. Relative performance metrics are calculated by normalizing the value with the ones obtained by the industrial design.

Another preprocessing step is a similarity analysis between the selected designs of each group and the components and actuators that have been simulated in OMOD. Two motors or gear pairs are considered similar if their decision variables are each within a 1×10^{-6} absolute or 1×10^{-4} relative tolerance. The similarity of two optimizations is calculated, if the searched actuator configuration is the same, by counting the number of entries of the definition file that are equal. Finally, the similarity of two actuators is calculated, for actuators sharing the same configuration, by counting the number of design variables that are within 1×10^{-3} absolute tolerance of each other. These similarity scores also allow linking actuator simulations and optimizations.

Finally, the different optimizations are classified into one of these eight categories (first fit):

1. **Example** if 80% or more similar to the example file;
2. **Error** if the definition contains unknown variables (mostly typo errors);
3. **Test** if the optimization budget is smaller than 10 000 or the configuration only includes two gear stages;
4. **Badly defined** when a constraint or an objective is incorrectly configured (e.g., minimizing compactness or requiring the resolution to be smaller than 6400);
5. **Missing constraint** when constraints related to the specifications or the chosen objectives are missing (e.g., impose a minimum torque excess unless it is an optimization objective);
6. **Invalid OP** when there are no 9 V supply condition or the total number of cycles is

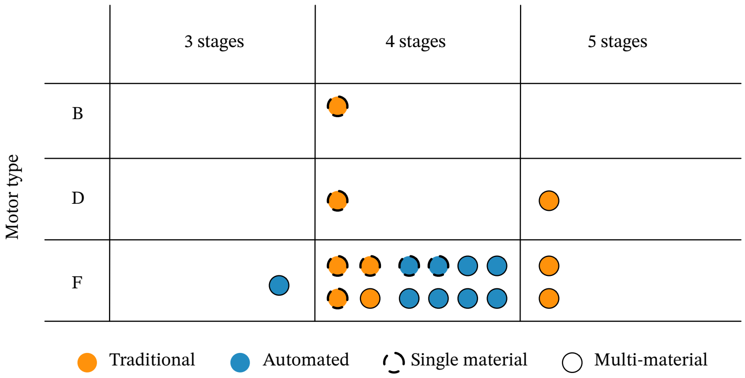


Figure 5.7: Comparison of the proposed actuators based on selected motor, number of stages and the use of multiple materials for the gears.

lower than 99 000 cycles for the chosen operating points;

7. **Constrained** if the optimization did not find any feasible solution;
8. **OK** otherwise.

In addition to this categorization, the sequences of optimization submissions is considered with the aim to identify batched submissions. In sequential submission, students submit an optimization, wait for the results, and submit a follow-up optimization based on the previous outcomes. In contrast, batched submissions are series of optimizations submitted by a group within a short time and that run in parallel, prohibiting feedback between the optimizations of the batch. Unfortunately, the dataset does not allow knowing precisely when an evaluation was completed. However, the chosen optimization budget allows estimating a lower bound for the computation time. Optimizations that are started before this estimate from the previous submission and at the latest within 15 min are considered to form a batch.

5.3.4 Impact on the selected products

Results

Starting with a high-level comparison, Figure 5.7 displays all the actuators from the students grouped by selected motor, number of stages and whether a single polymer is used for all gears. In the traditional approach, actuators have 4 or 5 gear stages and diverse motors, but about half are restricted to a single material. The actuators selected during the automated design phase use motor F only, have mostly 4 stages with a single 3-stage actuator and are composed of more different materials.

Moving to the quantitative analysis, Figure 5.8(A) shows paired plots comparing the actuators from the traditional to the automated phase based on four metrics related to the specifications. Looking at the torque excess, the security factor to gear tooth bending and

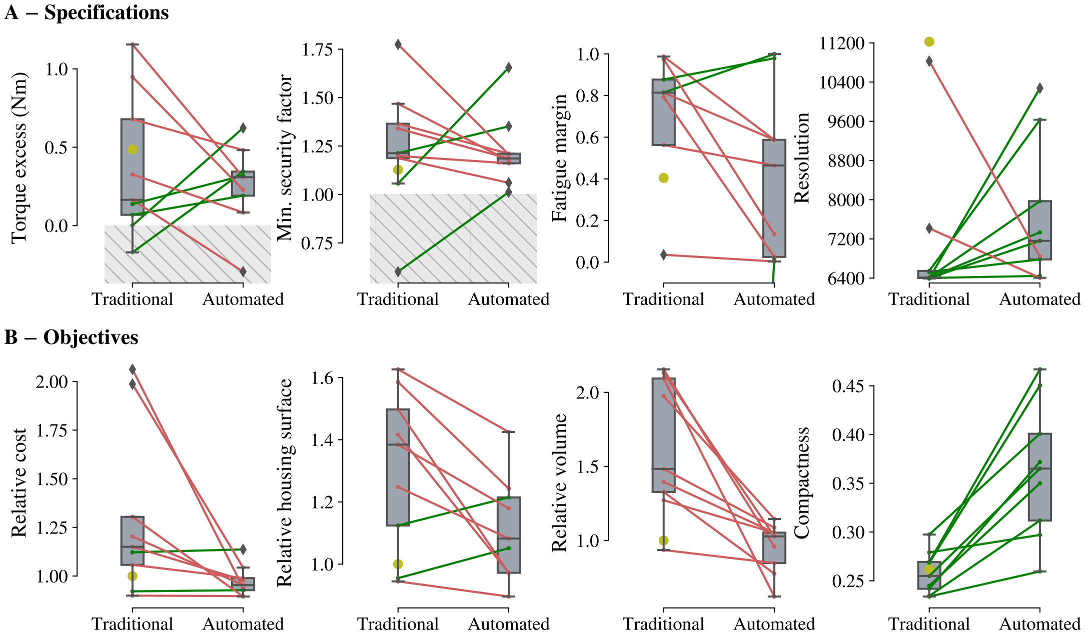


Figure 5.8: Paired plots comparing actuators between the two phases based on (A) specifications and (B) performance objectives. All individual solutions are represented as dots and colored in green and red to indicate an increase or a decrease from the traditional to the automated design respectively. Grayed zones indicate specification violation zones. The industrial product is shown as an olive point.

the fatigue margin, it appears that with the automated design tool, students have chosen solutions closer to the specifications, thus avoiding unnecessary overdesign. For the first two metrics, one also sees a reduction in the inter-group variability. On the contrary, a broader spectrum of resolutions—linked to the transmission ratio—is achieved by actuators from the automated design phase. In terms of objective metrics, Figure 5.8(B) shows paired plots for cost and three volume-based metrics. Seven out of nine groups have been able to reduce cost by using the automated design tool, while two groups have actuators with similar cost. The reduction of cost can be related to an important reduction of volume, an increase of compactness—fraction of useful volume in total volume—and a decrease of housing surface. With narrower margins to the specifications, the inter-group variability of cost and volume is similarly decreased. Overall, both figures suggest a more efficient use of material and volume in actuators originating from the automated design phase.

Figure 5.9 compares all actuators using two volume metrics against cost and further highlights that most groups were able to improve their product on multiple aspects simultaneously using the automated design suite. In addition, seven designs Pareto dominate the industrial product on cost and compactness, four of which also dominate it on volume. Figure 5.9 also highlights the diversity of the obtained designs.

It is worthwhile noting, however, that not all actuators respect the requested specifications.

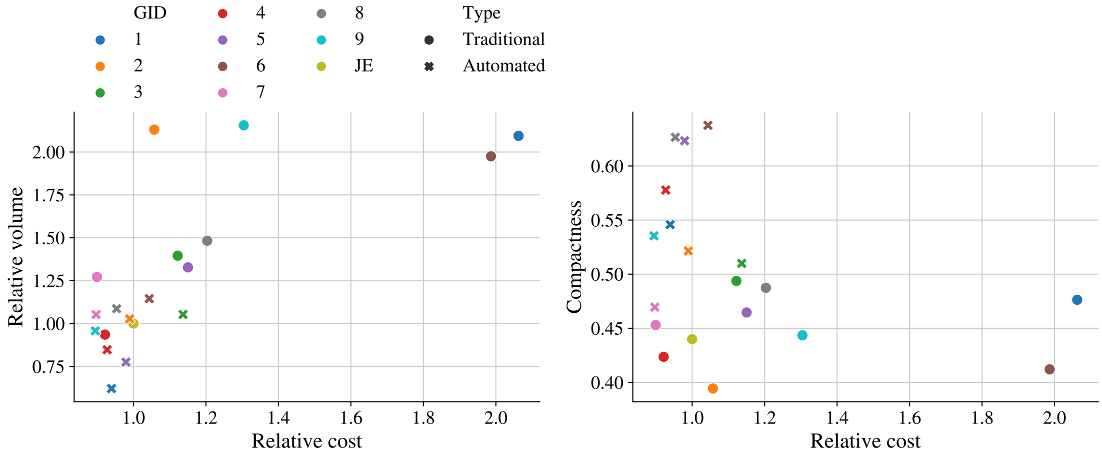


Figure 5.9: Biobjective comparison of the actuators where colors represent each group (GID) and marker styles each phase.

Violated specification	Traditional	Automated
Gear quality (gliding speed)	8	4
Min. torque & stresses	4	6
Packaging (assembly, collisions)	2, 7, 9	-

Table 5.11: GID of the groups whose design violates certain specifications.

This is the case of five traditional designs and 2 automated designs. Table 5.11 summarizes the violations. The number of issues with packaging is surprising since two of the three groups have chosen the simpler step-like stacking of gears—each gear pair is put on top of the previous one. While automated design did not help resolve all issues, even if it can handle them, it did eliminate packaging related issues.

Improvements in packaging are, however, not the only reason for better use of space. Figure 5.10 shows the diameters of pinion and wheel, and the thickness of the gears per stage. Compared to the gears designed by the students, the automated design tool suggested:

1. more differentiated wheels depending on the stage;
2. smaller pinions across all stages;
3. thinner gears up until the output stage.

In addition, the diameters of the pinions and wheels selected by the automated tool are closer to those of the industry product. The reduction in pinion size is possible, because contrary to most student groups, the automated design tool leverages the profile shifts of the gears to use pinions with down to nine teeth. Indeed, five out of nine groups have used a single pair of profile shifts for all stages, three of which have not even followed the recom-

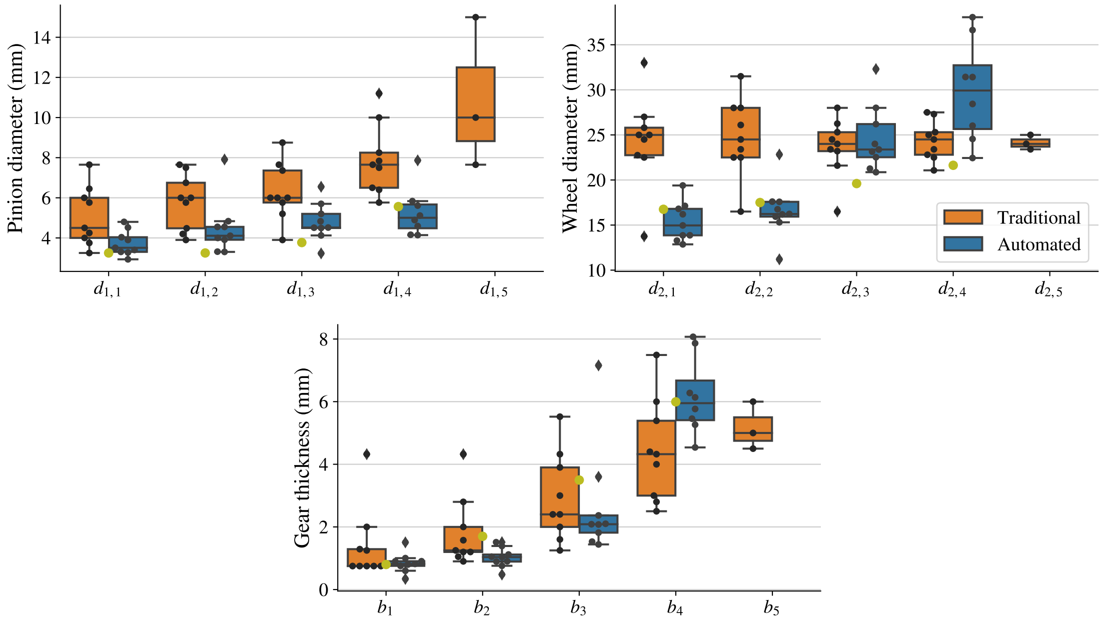


Figure 5.10: Comparison of three gear parameters by stages between traditional and automated design. Individual points are represented by dots. Since not all designs have the same number of stages, the number of points varies by stage. The parameters of the gears of the industrial product are shown in olive.

mended long and short addendum—or "V-Null"—rule [128], suggesting that the students were not familiar enough with the concept of profile shift. In such conditions, it is difficult to design pinions with less than 16 teeth.

Discussion

Considering the products, the use of an automated design tool has enabled students to find designs that are less overdesigned and that make better use of the material and available space. The obtained actuators are also closer to the industrial product in terms of selected motor and gear design, which is not necessarily a surprise since the tool has been developed for that purpose. Particularly striking however, is that some groups have been able to suggest actuators that outperform the industrial product on several metrics simultaneously.

While purely in terms of performance, the actuators designed by students during the traditional phase are fully competitive, the fact that a majority violates the required specifications highlights the difficulty of this design task. Indeed, the design of a 4-stage actuator is a coupled parameter design problem with 35 variables requiring system-level awareness, something humans are known to struggle with [199]. The typical trial-and-error approached followed by novice engineers [3] due to their lack of domain knowledge [19] is in such cases very time-consuming. The selection of the motor and the gearbox are interdependent and have a significant influence on the packaging. Further, several specifications concern the whole system, which thus needs to be completely defined in order for these specifications to be assessed.

It is therefore interesting to look at the strategies used by the students and their effects on the product. In the traditional segregated design approach, one would typically decouple the motor from the gearbox by choosing *a priori* the transmission ratio of the gearbox. The minimum angular output resolution sets, with that respect, a lower threshold on the transmission ratio and most students (7/9 groups) seem to have followed this path, leading to actuators all very close to the required resolution. In addition, the following simplifications made by the students are visible in their design:

1. Limited pinion designs (one or two different numbers of teeth), and large and safe number of teeth which avoids the need to use profile shifts;
2. "Constant" wheel sizes across stages, allowing easier packaging;
3. Restricted choice of materials (5/9 use a single material);
4. "Step-like" gear layout.

Most simplifications hinder however to find a truly optimal design and certain design decisions can seem counter-intuitive. This explains the large conceptual changes in the actuators recommended by the students from one approach to the other. As an example, choosing the smallest allowed output resolution might seem to be an adequate decision when trying to obtain smaller systems. The actuators obtained through optimization are more compact, while having a greater output resolution, providing a counter-example to this intuition.

5.3.5 Understanding the process

The traces left by the students in OMOD offer more information than just a snapshot of the final product. They suggest the design path taken by them and this section analyzes and discusses what the logged evaluations can tell and what not. To get a first sense of the amount of work done by the students, Table 5.12 summarizes the number of requests made by each student. With a total of 4792 requests in the first phase and 1218 requests in the second one—of which 686 are optimizations—it seems clear that the platform has been extensively used.

Traditional design phase

The numbers in Table 5.12 seem to confirm a decoupling between motor and gears, since in most groups, students seem to have simulated either one or the other. Additionally, the students seem to have put, comparatively, more efforts into numerically exploring the motors than the gears. Indeed, excluding group 8 which accounts for 60% of the evaluated gears,

Table 5.12: Number of unique simulations by available model per student.

GID	UID	Motor	Gears	Actuators	Optimizations
1	5	16	87	116	18
	10	20	7	18	3
	30	16	6	3	1
2	1	0	114	23	36
	8	106	0	5	62
	11	38	0	0	53
	31	1	73	0	51
3	12	21	130	1	23
	24	31	12	4	7
	25	11	117	57	22
	27	10	33	0	0
4	6	0	29	0	3
	17	8	100	11	10
	23	65	0	0	7
5	3	10	131	5	6
	4	52	8	9	25
	15	64	46	65	48
	16	15	0	2	13
6	2	0	81	22	14
	13	0	0	0	0
	18	64	69	6	64
7	21	29	68	21	55
	22	15	82	6	5
	26	2	100	12	30
	33	0	58	0	53
8	7	1	3	0	0
	9	6	0	26	15
	19	16	46	67	33
	29	11	2275	0	0
9	14	68	0	34	27
	20	101	0	16	2
	28	79	122	2	0
	32	119	0	1	0
Total		995	3797	532	686
Average		30	115	16	21

there are 961 motor against 1473 gear simulations, whereas in terms of complexity, there are three variables for motors against eight per gear stage. Nevertheless, all groups seem to have followed a data-driven exploration approach. Indeed, a singular value decomposition—also called generalized eigendecomposition—of the matrix with all the simulated variants returns a full or almost full rank for all groups, suggesting that the influence of each variable has been evaluated to some level.

Figure 5.11 shows the spread in time of the requests for motor and gear simulations during the traditional design phase. It highlights the different approaches followed by each group. The following trends can be noted:

1. The decoupling between motor and gearbox is also temporal for many groups.
2. Most groups started by exploring the motor (7 of 9 groups).
3. On average students spent less time on gears (20 d 7.56 h, SD 15 d 6.7 h) than on the motor (37 d 7.36 h, SD 13 d 19.1 h).
4. Groups 4, 6 and 8 did not simulate all selected stages—group 9 uses the same gear design for all stages.

Within the gearbox, some groups seem to go from the motor to the output, e.g., group 2 or 3, while others go the other way around, e.g., group 6. Overall, however, it is difficult to offer conclusive statements, since it is not possible to know whether a component simulation is part of a general search process or the result of directed refinement. Especially for the gears, since some calculations could have happened outside the platform, it is difficult to interpret the intentions of the students.

Besides, an analysis trying to identify the stage to which each gear simulation belonged was not successful. In fact, the analysis revealed numerous inconsistencies in the defined input speed, torque, and number of fatigue cycles that hinder the classification of the simulations with sufficient confidence.

Studying the chosen operating points for their simulations also reveals that three groups (5, 8, and 9) have not performed, in the platform, the required simulations to ensure their gears fulfill the specifications. Three groups (4, 6, and 8) have not simulated the performance of their chosen motor in worst-case conditions (9 V supply and extreme temperatures). Lack of validation does not, however, imply that the actuators do not meet the specifications. Indeed, only group 4 ended up having too little torque and failing the fatigue specifications—although this group performed the simulations confirming they would not pass fatigue.

To summarize, students executed numerous simulations certainly as a way to understand how the models they were given worked. In particular, the comparatively large number of simulations and the longer duration of the search for the motor—which is not the primary domain of mechanical engineering students—are inline with the findings of trial-and-error

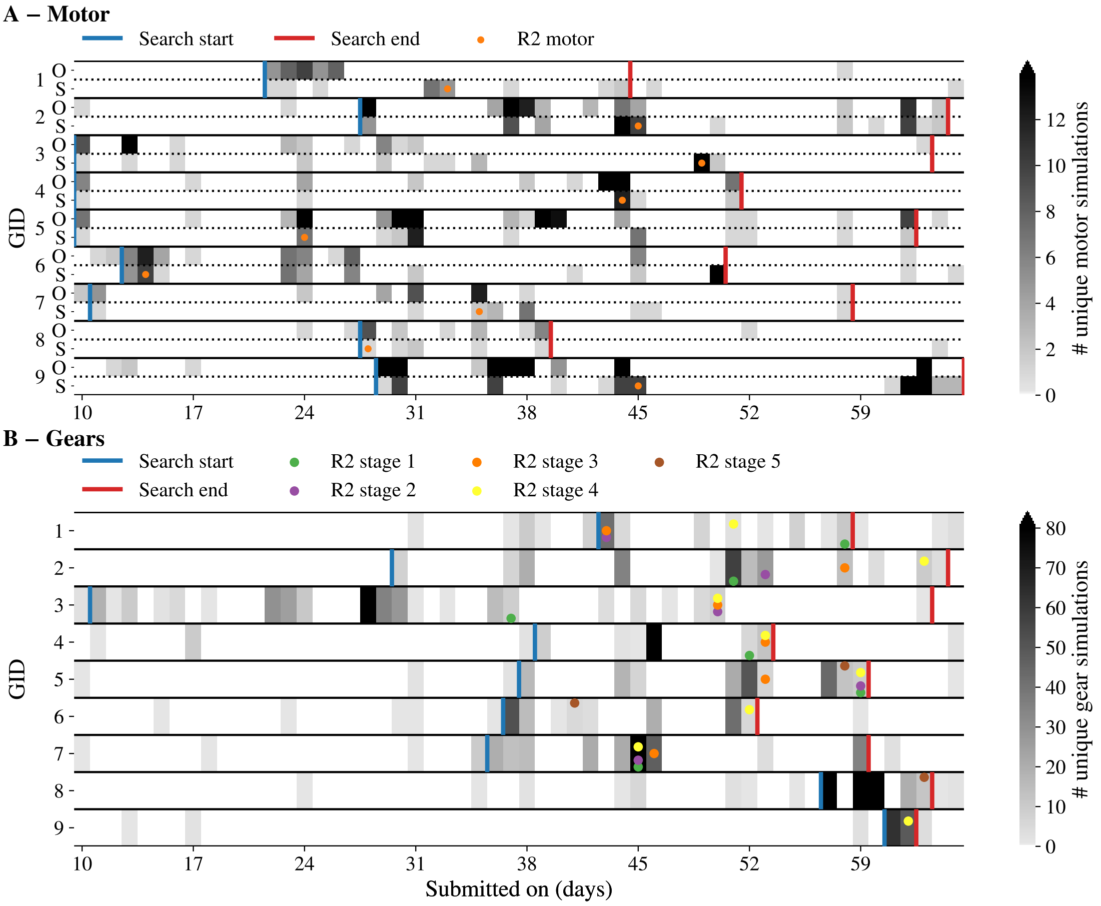


Figure 5.11: Heat map of the number of simulations per day and per group. The shown time window starts from the start of OMOD to report 2. The colorbars are cropped at the 98th (A) and 99th (B) percentiles. The ticks of the x-axis correspond to the days on which the weekly lectures or meetings were scheduled. The blue and red bars indicate for each group the start and the end of the search process, meaning that there are more than 4 (A) or 5 (B) consecutive simulations with less than an hour in between each. (A) displays the motor simulations, and for each group distinguishes between the simulations of the same motor kind "S" and the others "O". The orange dots show the first time the exact motor configuration selected in report 2 is evaluated. (B) displays the gear simulations. The colored dots show the first time a stage of the actuator of report 2 is simulated.

Table 5.13: Number of optimizations by category (see Section 5.3.3).

GID	Ex.	Error	Test	Badly def.	Missing c.	Invalid OP	Constrained	OK
1	3	1	4	10	0	4	0	0
2	5	2	68	6	59	17	7	41
3	2	3	15	32	0	0	0	0
4	1	0	8	0	8	0	0	3
5	1	10	55	11	10	0	1	5
6	5	0	26	45	3	5	2	0
7	1	1	107	7	0	0	6	31
8	12	0	9	11	3	16	1	0
9	2	2	11	0	0	0	0	15
Total	32	19	303	122	83	42	17	95

approaches by novice engineers [3]. These results also echo the findings that the use of prototypes—and numerical models can be seen as virtual prototypes—by novice engineers is often unintentional [43]. Under such conditions, it is difficult to "optimize" solutions within the allotted time.

Automated design phase

The start of the automated design phases means students get their hands on an integrated model for actuators and an optimization module. Based on the categorization method, Table 5.13 lists the number of optimizations by category performed by each group. Overall, the largest share of optimizations falls into the test category. Since the computational resources were limited, students had been asked to perform optimizations with little budget until they were more confident about their configurations. It also appears that all groups have run at least once the example file. Then, depending on the groups, different behaviors and approaches can be seen: few, but mostly well-configured optimizations—e.g., group 9 with 50% OK—many tests like group 5 or 7, and everything in between. Group 3 did not perform any simulation past the badly defined category, and groups 1, 6, and 8 have no optimization using the worst-case operating points. Translated in terms of total computational budget, the groups have evaluated for all their optimizations between 527 000 and 7 392 520 solutions.

In terms of evolution, Figure 5.12 shows a heat map of the number of optimizations performed by each group per number of stages. The map shows that most groups started their optimizations about a week after the beginning of the second phase and stopped few days before their final oral presentation. Two groups performed additional optimizations into the Christmas break (W15 and W16). On average, students spent only 20 d 15.8 h (SD 7 d 14.1 h) on optimizations, but as for the traditional design phase, there are various behaviors. As had been recommended by the teachers, six groups have rapidly implemented their actu-

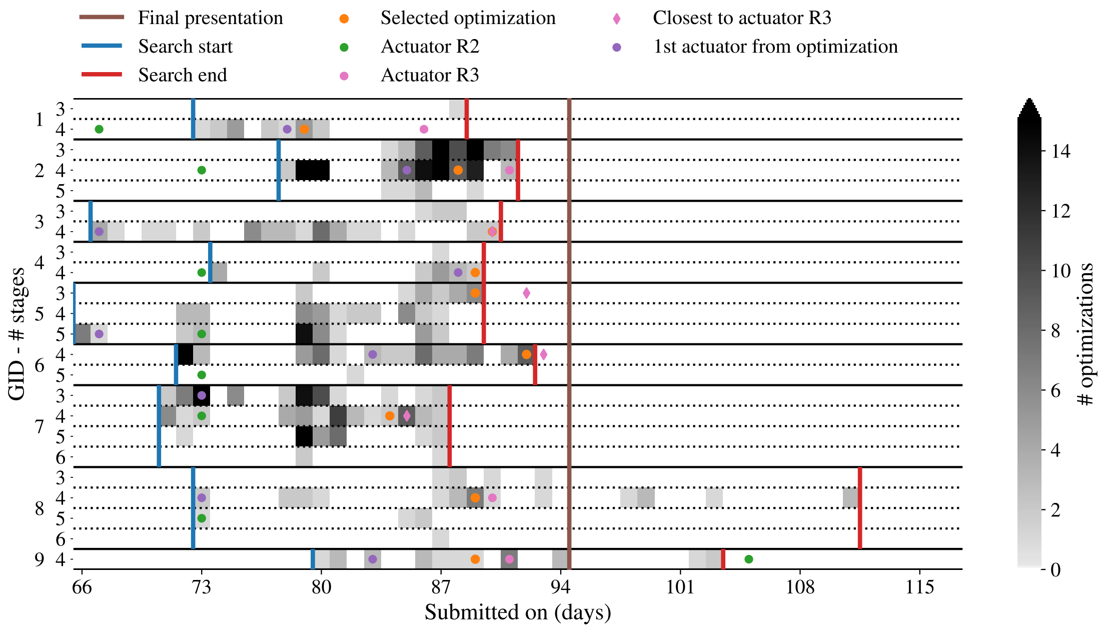


Figure 5.12: Heat map of the number of submitted optimizations per day, per group and per number of gear stages (showing only ≥ 3). The shown time window spans the automated design phase. The oral presentation is shown as a brown bar. The colorbar is cropped at the 98th percentiles. The ticks of the x-axis correspond to the days on which the weekly lectures or meetings were scheduled. The blue and red bars mark the first and last day when optimizations that are not the provided example are executed. The orange dots show when the optimization from which the final design stems is performed. The pink dots and diamonds show when the actuator in or the closest to report 3 has been simulated in the actuator module (first appearance). The green and purple dots indicate when the actuator of report 2 and the first actuator linked to an optimization has been simulated in the actuator module.

ator from report 2 into OMOD to help them understand the way the integrated model works. In addition, five groups have started to test obtained actuators in the actuator module early in their exploration process.

Another interesting way to look at this mass of optimizations is to count the number of different inputs the students gave for the various settings, see Table 5.14. The numbers highlight the exploratory behavior followed by most groups. Specifically, students have investigated many objective combinations, constraints, but also, more surprisingly, many search bounds for the gears. There are not many settings that students have not tried to change. Table 5.15 shows the number of times optimization with the same set of settings—excluding optimization configuration like the budget or the bounds—have been performed, which further confirms the significant exploration effort done by students. Most optimization configurations have been run only once and few have been run more times, probably to try longer optimizations or to adjust the number of stages searched for.

Table 5.14: Number of different settings evaluated by students in their optimizations, excluding example optimizations.

GID	# stages	OP set	Objectives	Constraints	Motor bounds	Gear bounds
1	2	6	5	11	3	4
2	3	15	44	60	9	31
3	3	4	11	18	4	13
4	2	2	7	10	1	4
5	4	9	40	43	8	18
6	2	3	25	24	8	5
7	5	5	31	21	11	9
8	5	3	16	18	8	11
9	2	3	9	18	5	4

Table 5.15: Number of times similar optimizations—differing solely in terms of optimization settings (budget or search bounds)—have been performed.

GID	1	2	3	4	≥ 5
1	14	2	0	1	0
2	72	40	11	2	1
3	20	7	0	1	2
4	13	2	1	0	0
5	70	3	1	1	2
6	39	10	3	3	1
7	40	12	8	4	6
8	23	7	5	0	0
9	18	6	0	0	0

Considering the class as a whole, Figures 5.13(A) and (B) show two patterns that develop over time. Figure 5.13(A) distinguishes between batched and non-batched optimizations and shows that while mostly absent initially, batched optimizations are gradually increasing until they become dominant in the last week before the oral presentation. Once one knows how to use the automated design tool, batched submissions are the most effective to evaluate multiple different possible designs. Figure 5.13(B) considers the similarity to the example and shows how the submission by the students gradually diverge from it. It seems students have performed numerous small steps away from the example to experiment with their effects until they found the settings they thought were most appropriate. Figure 5.13(C) shows the validity with respect to the specifications of the simulated actuators originating from an optimization. While at first, no valid actuator was simulated, the quality of the obtained actuators seems to increase. The search for valid actuators peeks following the last course and seems to suggest a certain learning process.

However, Figure 5.13(C) also shows that throughout the automated design phase, there are an approximately constant amount of simulated actuators that do not produce sufficient torque for the application. They represent 40% of the simulated actuators and the lack of torque is the violation in 75% of the cases. It seems peculiar that students did not correct their optimizations and it is questionable whether the students did realize this problem. Looking at the set of operating points for these actuators reveals that their performance are mostly verified at 12 V or higher. Compared to the other actuators ($N = 182$), the actuators lacking torque ($N = 130$) have been simulated with a significantly lower running torque requirement ($p < 0.001$, two-sided Mann-Whitney U test [119]) and a significantly higher supply voltage ($p < 0.001$, two-sided Mann-Whitney U test). This difference in the chosen supply voltage for the operating points can also be traced back to the linked optimizations ($p < 0.001$, two-sided Mann-Whitney U test). Thus, the incorrect identification of the worst-case conditions to guarantee the specifications is the key reason behind invalid actuators. The results also suggest students might not have been aware of it, and the teachers missed pointing this out.

Nonetheless, there has been a total of 138 valid actuators simulated by the students. Figure 5.14 displays them in a relative cost and relative volume map, and reveals the great diversity of the solutions found by the students. In particular, an impressive 49 actuators are below the $(1, 1)$ threshold, meaning they outperform the existing industrial product on both metrics. Yet, not all groups have contributed to these solutions, and groups 7 and 9 account for 43 of those solutions.

The relations between the optimizations from which the selected designs have been chosen and the products described in report 3 are shown in Table 5.16. Considering the optimizations, most groups have restricted themselves to biobjective optimizations and all groups haven chosen to minimize cost along with a volume-related metric. The chosen optimizations fall into the OK (4x), invalid OP (3x) and missing constraint (2x) categories. The invalid actuators are related to a missing constraint and an invalid OP optimization. Yet, not

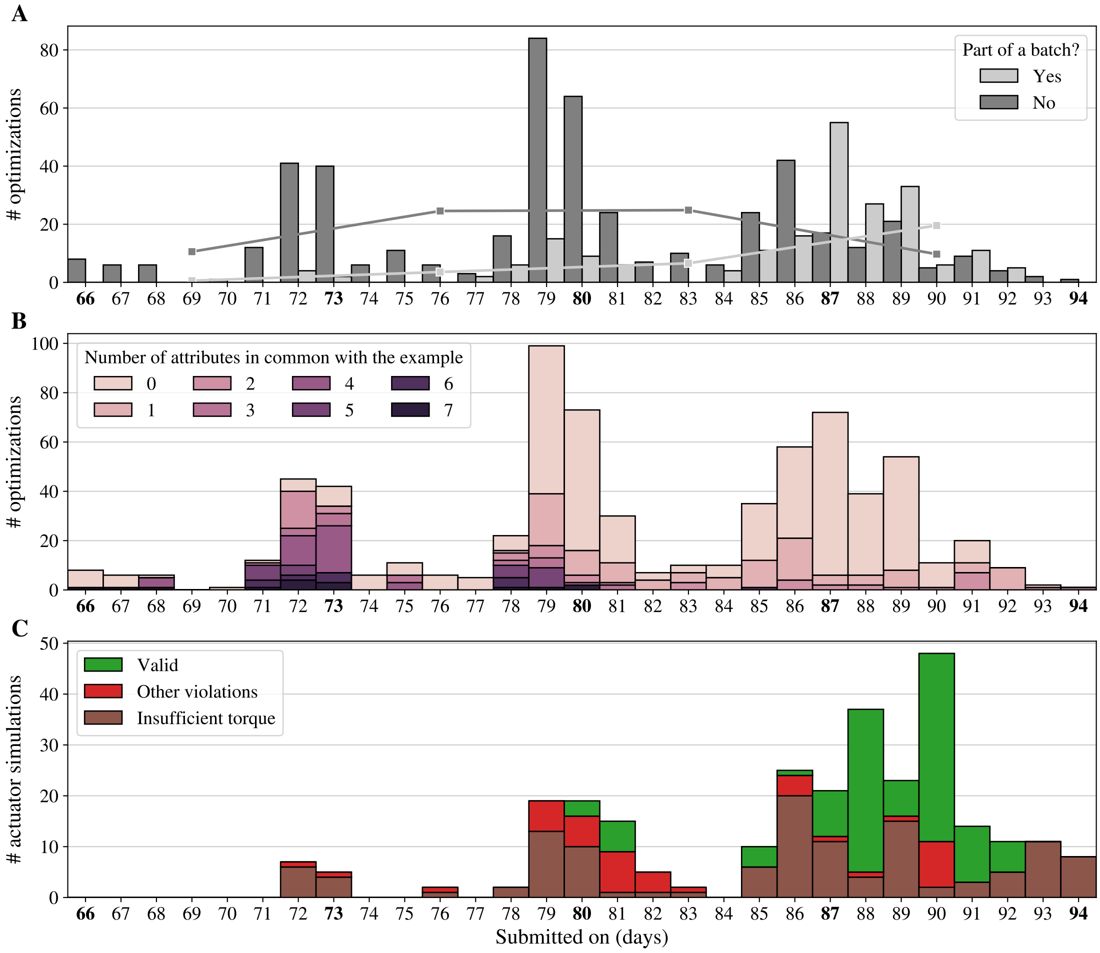


Figure 5.13: Per day view between the start of the automated design phase and the final presentation. (A) Number of optimizations separated based on their inclusion in a batch. The daily means for each week are shown as a solid line. (B) Number of optimizations colored by the number of attributes (e.g., chosen objective combinations or constraints) they share with the given example file. (C) Number of actuator simulations originating from an optimization colored depending on how they respect the specifications.

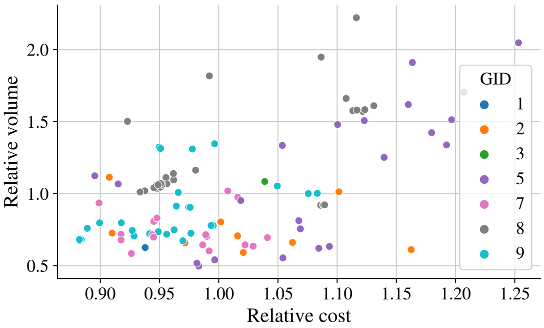


Figure 5.14: Map of all valid actuators originating from an optimization based on their relative cost and relative volume, and colored by group (GID).

Table 5.16: Overview of the optimizations from which the actuators in report 3 have been chosen and the differences.

GID	Objectives	Budget	Opt. category	Differences to opt.	Simulated R3	Valid
1	Min. cost Min. assembly steps	60 000	Invalid OP	d_3, d_4	Yes	Yes
2	Min. cost Max. safety factor Min. volume	48 000	OK	Exact	Yes	Yes
3	Min. cost Min. volume	150 000	Missing constraint	$FF, R_{scale}, d_1, d_2, d_3, d_4$	No	Yes
4	Min. cost Min. thickness	100 000	Missing constraint	Exact	No	No
5	Min. cost Max. compactness Min. volumne	150 000	OK	d_1, d_2, d_3	No	Yes
6	Min. cost Max. compactness	148 000	Invalid OP	$d_1, d_2, d_3, d_4, \gamma_3$	No	No
7	Min. cost Min. volume	144 000	OK	d_1, d_2, d_3, d_4	No	Yes
8	Min. cost Max. safety factor	150 000	Invalid OP	$R_{scale}, \gamma_1, \gamma_4, b_2, b_3, b_4$	Yes	Yes
9	Min. cost Min. length Min. thickness	50 000	OK	Exact	Yes	Yes

all actuators selected from such optimizations are invalid and adjustments made by students of group 3 and 8, for example, have allowed their design to pass the specifications. In fact, a majority of groups have decided to manually adjust—beyond simple rounding—the optimized design, mostly to increase inter-component spacing. As a matter of fact, the automated design tool available to the students could return unrealistically small spacing ($< 1\,\mu$m). However, only two out of six groups investigated the impact of their changes on the predicted quality, as seen by model. Finally, there is no correlation between the cost of the actuator and the budget of the associated optimization, e.g., group 9 has the lowest cost and the second lowest budget.

5.3.6 Discussion

In general, students have proven to be remarkably curious and dedicated to mastering the various tools. Not yet conditioned by experience and fearful of a trial-and-error approach, students have tried to understand the mechanics behind all modules following a black-box-like exploration. In the context of automated design tools, where structured processes and guidelines are not established [153, 190], this approach has allowed them to learn and apply a novel tool and method unknown to them within on average 20 d 15.8 h (SD 7 d 14.1 h). Doing so, they were able to suggest actuators that were better than the ones that they worked on for an average 42 d 1.56 h (SD 7 d 8.07 h), and even better than the industrial product this application is based on. As an extension, they confirmed the validity and usefulness of the

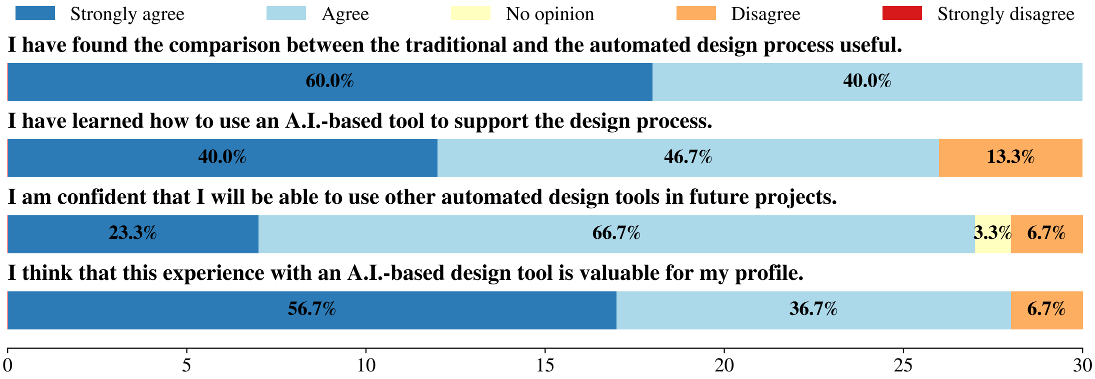


Figure 5.15: Anonymous student feedback for the MA course ($N = 30$).

developed automated design framework.

The integrated model is essential to tackle system design and allowed students to reduce the packaging-related violations of their design. It offers a real support to better include system-level constraints, a challenge for most humans [199]. It did, however, not help to address other violations, among which the lack of torque. This could be related to an erroneous identification of the operating points to input. While some misconfigurations could and should be automatically detected—it might have saved the students time if they had received a warning about all the badly defined optimizations—not all can be inferred by the tool. Human engineers still have to translate customer requirements into engineering specifications, and then into “computer specifications”, a challenge common to many engineering software. This clearly suggests that stronger problem analysis and computational thinking skills should be promoted in engineering education.

When thinking of artificial intelligence, people often associate it with expensive computer hardware requirements. While “Big Data” approaches often need large infrastructure,s some automated design tools, such as the one used here, do not necessarily need such investments. OMOD and its workers were running on a single 6-core machine that could handle the workload of nine teams. Including queuing and optimization time, most results were available within 1.46 h (excluding example and test optimizations, IQR [0.33 h, 4.19 h]), which is reasonable for the obtained advantages.

Finally, out of all these technical considerations, the feedback of students regarding the course they attended is shown in Figure 5.15. Overall, the students were extremely satisfied by the content of the course, and they mostly feel they learned from the experience. In particular, they think they will be able to use this experience in their future occupation, although given the answers, it seems they don’t know yet how this transfer will exactly happen. Here, while this only reflects the students’ beliefs, a similar question remains if the demonstrated correlation between self-efficacy beliefs and performance [203] holds and if this will indeed promote the use of machine intelligence driven engineering tool in industry.

Limitations

The decision to perform the study in real conditions and in a teaching setting conditions the study design. The size of the studied population may be limited but corresponds to the traditional size of this class. The behavior of students is not predictable and is subject to external and unknown disturbances and motivations. For example, the collected time sheets show large deviations in the reported time worked on the project: $[59.5, 272.25]$ h, and with the data at hand, it is impossible to explain whether these differences are real or not. Yet, this setting makes it realistic, because much of this also happens in industry, even if not many would admit it.

Despite the best intentions, this study design cannot answer all questions. Many students argue, for example, that having performed the design task by hand help them learn the automated design tool. In addition, the developed platform did not gather data on the interactions of the students with the results or ask students to state their intention when submitting evaluations. These are many important points to investigate to further develop the use and the education of machine intelligence in mechanical engineering.

5.4 Concluding Remarks

In this chapter, the results from two different investigations performed with students have been reported. They yield valuable outcomes regarding the education of future engineers, as well as the development of design automation tools.

In terms of skills, the results suggest that students can develop their professional skills through projects, but that appropriate support and feedback are needed to strengthen their learning, like for any other skill. Further, the results show that when technical computations are handed over to computers, the importance of problem-solving and computational-thinking skills becomes more apparent. Indeed, the inappropriate selection of the operating points in the tool was related to the generation of invalid designs. So, educating future engineers also means highlighting to students those challenges.

Through their use of the automated design tool, students were able to:

1. propose more qualitative designs, both in terms of performance metrics—e.g., cost or volume—and specification violations;
2. get rid off packaging errors;
3. simulate 49 actuators that outperform the existing industrial product in terms of cost and volume;
4. investigate several trade-offs.

Furthermore, they did all of this on average in just over 20 days, including learning time.

Given that students also had other courses in parallel, this performance is certainly unprecedented for novice engineers. With most optimization results available within 1.46 h, students could experience the fast iterations and decision-making loops envisioned in the introduction of this work in Figure 1.3.

Et maintenant,
Maître, c'est moi qui te convie
À vider cette coupe où fume en bouillonnant
Non plus la mort, non plus le poison; mais la vie!

Méphistophélès (Faust, C. Gounod)

6

Conclusion

DESIGN AUTOMATION is a central concept to tackle the design of complex mechanical systems. Within the framework of the design of electro-mechanical actuators, three important ingredients were investigated: integrated modelling, efficient constrained multiobjective optimization, and future-looking education of engineers. Through this work, like pieces of a puzzle, as the shortcomings are addressed, these ingredients come together to build an effective automated design tool.

6.1 Summary

The developed numerical model of actuators uses a modular approach to build up a system view starting from interconnected components, such that different configurations can be simulated. The focus was put on fast component models to free resources for modelling complex system-level constraints, which are close to typical specifications. At the component level, a steady-state model for stepper motors was derived from a set of ordinary differential equations (ODEs), while standard norms were applied for gears. At the system-level, the layout of the actuator was considered and a 3-D representation of each system was built. Leveraging fast techniques developed for the computer graphics industry, packaging and assembly constraints were modelled.

The multiobjective optimization problems to obtain optimal actuators proved hard to solve by state-of-the-art algorithms. This was highlighted by proposing a realistic benchmark framework: Multi-Objective Design of Actuators (MODAct). This is one of few other attempts to provide the evolutionary computation—and the broader operational research—community with challenging optimization problems that better capture the properties of problems from real applications. MODAct offers a standardized way for anyone to test the performance of their optimization algorithms on mechanical design problems without needing domain knowledge. The constraint landscapesd of the 20 new problems were characterized using existing metrics of the decision space—feasability ratio (FsR) and ratio of feasible boundary crossing ($\mathrm{RFB}_{\times}$)—as well as newly suggested metrics—PFd and PFcv—

aiming at quantifying the effect of constraints in the objective space. This analysis and a convergence study using NSGA-II, NSGA-III and C-TAEA confirmed differences between MODAct and benchmark problems from literature.

Building on this new test suite, a large investigation of relaxation-based constraint handling strategies (CHSs) was performed. The investigation has, once more, shown that MODAct problems are different to many existing benchmark problems. The novel cEpsilon CHS, introduced in this work to tackle many-constraint problems, proved to be the best compromise across all problems, with in particular, outstanding performance on mechanical design problems.

Based on the integrated model and multiobjective optimization, a tool for the automated design of actuators was created. Its qualities and potential were highlighted through two case studies by the author, and more importantly, by the outstanding designs, which students—i.e., novice engineers—were able to generate within a short time. Both author and students found numerous design alternatives that show improvements over existing and established industrial products. And while students were not always able to satisfy packaging specifications when designing "by hand", the system-level constraints of the automated design tool eliminated this iterative and tedious task. These constraints were also shown to be key to ensure that qualitative and realistic solutions were generated by the tool. Further, the generation of 3-D representations allowed a rapid transfer from numerical vectors to tangible concepts. All in all, the promise of fast iteration loops of design automation is fulfilled.

The increasing automation and digitalization are favoring professional and computational-thinking skills. Two studies with the participation of students were performed to understand how to better shape the future of engineering education. On the one hand, the study suggests that students' learning of professional skills through team-based project is limited in general, but can be supported by explicitly teaching those skills and by providing feedback to students about them. These results suggest that promoting the development of such skills by students does not require a full curriculum redesign, and that instruments for assessing professional skills even in large classes exist. On the other hand, students were asked to design actuators for an automotive HVAC application using a traditional approach, followed by an automated design approach. The students were given access to the developed numerical models through a web interface, allowing their requests to be stored. The students learned how to use the platform and the models by performing black-box-like searches. They put significant effort in understanding the effects of the many parameters, and their efforts paid off. This study showed, in particular, that once the technical challenges related to the design task were taken care of by the automated design tool, only problem-solving and computational-thinking related questions remained to be supported by educators. Through this study, students had a unique opportunity to experience design automation and its process. Satisfied with their experience, they will carry it over into their professional careers.

Finally, handing over the automated design tool to many students also contributed to improving the tool itself. Not only were bugs identified and corrected, the learning path followed by students emphasized the importance of early error detection and rapid feedback. These two key characteristics should be part of any developed tool. In addition, contrary to a widespread belief, not all machine intelligence applications require extensive and expensive hardware.

6.2 Outlook

Several contributions of this thesis form the basis for future investigations:

1. The automatic selection of the most interesting actuator configurations is a valuable extension to the framework presented in Chapter 2. The optimization of variable-length decision vectors is a challenge that comes on top of proper constraint handling. The methodology, however, is of interest both for the design of many mechanical systems, for example high-speed turbocompressors, but also in other fields, such the configuration of neural networks. Adapted and meaningful search operators are one way to approach the optimization problem. The topic also questions the validity of the calculated performance metrics with different configurations and asks for strategies to include subjective configuration preferences that might be hard to capture numerically.

2. Further numerical assessment methods to better identify and quantify the properties of constrained multiobjective optimization problems should be developed. The motivation for such metrics is both practical and theoretical. In practice, they could be used to select the most appropriate optimization methods for a given problem, or to develop adaptive algorithms that require less tuning. They could also participate in transferring the properties of MODAct problems back into better "synthetic" problems that are useful for advancing theoretical research in the field.

3. There are still many open aspects of constrained multiobjective optimization to be investigated. MODAct offers the tools required to develop algorithms that are specifically targeted at similar applications and most importantly, that consider the constraints as a key part. Hybrid and ensemble techniques seem promising in that context, but the field may benefit from opening up to fresh ideas.

4. The level of knowledge and skills needed to use *intelligent* tools is a very important open question in the field and should be further investigated. Since in the presented study, students gained experience by solving the same design task using conventional methods before using the automated design tool, the question remains if prior experience supports the use of such tools.

Finally, as a general goal, researchers-educators should thrive to promote evidence-based education and think about how their research impacts the engineers and scientists of tomorrow.

A

Comparison of the steady-state stepper motor model

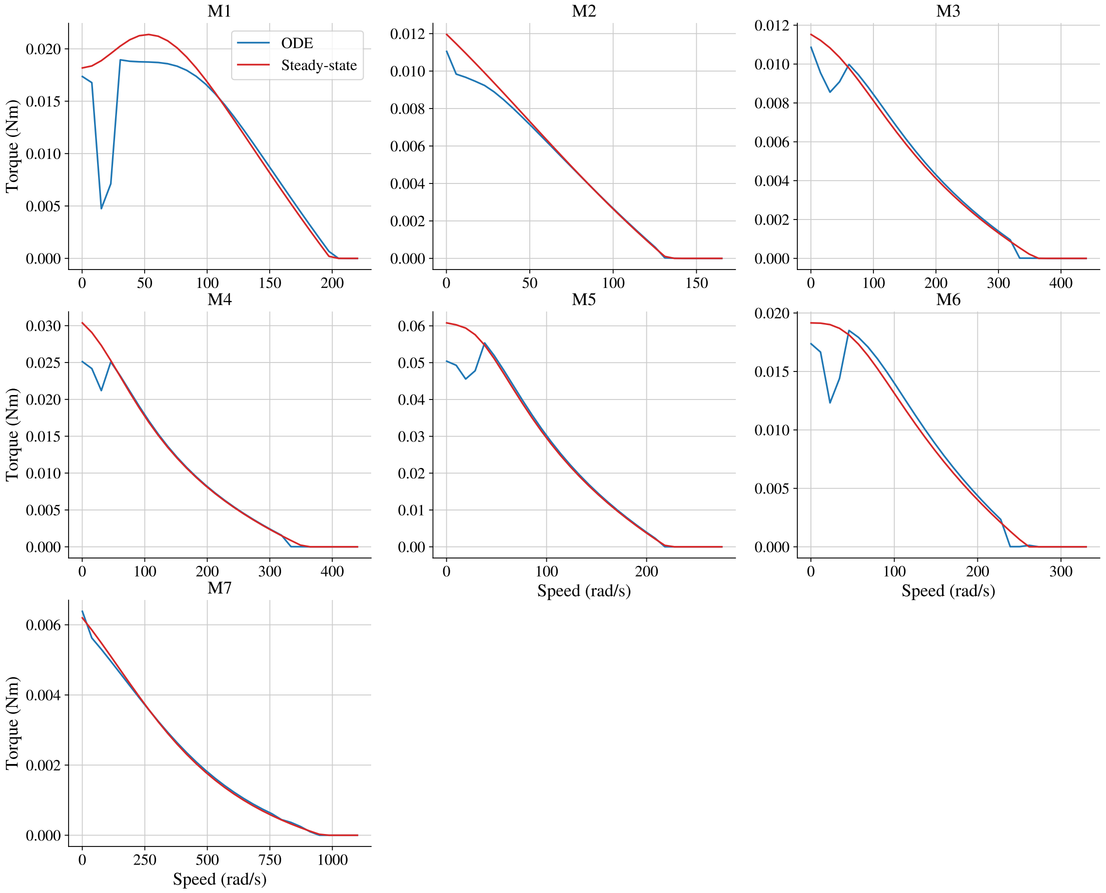


Figure A.1: Comparison between stepper models for all defined steppers at $V_s = 12\,\text{V}$ and with no current limitation.

B

Interprofessional project management questionnaire (IPMQ)

The questions of the Interprofessional Project Management Questionnaire (IPMQ) [99] are reproduced here to ease the understanding, with the permission of its authors.

Table B.1: List of the questions of the IPMQ in English and *in French* along with the factors they are linked to (A=planning, B=risk assessment, C=ethical sensitivity, D=communication, E=interprofessional competence).

	Questions	Factors
Q1	I am good at making a clear problem statement to clarify the goals when I start working on a project. *Je suis compétent.e pour formuler un énoncé clair du problème afin de clarifier les objectifs lorsque je commence à travailler sur un projet.*	A
Q2	I am good at defining a clear work plan early in a project. *Je suis compétent.e pour définir un plan de travail clair dès le début d'un projet.*	A
Q3	I am good at breaking a large project into a number of smaller work packages. *Je suis compétent.e pour diviser un grand projet en plusieurs petits lots de travail.*	A
Q4	I am good at analyzing a project work plan to identify the order, priority and importance of work tasks. *Je suis compétent.e pour analyser un plan de travail de projet afin d'identifier l'ordre, la priorité et l'importance des tâches à accomplir.*	A
Q5	I am good at identifying how to keep track of which tasks have been completed and how a project is progressing. *Je suis compétent.e pour identifier comment suivre les tâches qui ont été accomplies et comment un projet se déroule.*	A
Q6	I am good at clarifying how likely it is that something will go wrong with a project. *Je suis compétent.e pour clarifier la probabilité que quelque chose tourne mal dans un projet.*	B

Table B.1: (continued)

	Questions	Factors
Q7	I am good at identifying how much damage or trouble may be caused by something going wrong with a project. *Je suis compétent.e pour identifier les dégâts ou les problèmes qui peuvent se produire si quelque chose ne va pas bien avec un projet.*	B
Q8	When working on a project, I am good at estimating the likelihood and potential impact of something going wrong with a project. *Je suis compétent.e pour estimer la probabilité et l'impact potentiel d'un problème lié au projet sur lequel je travaille.*	B
Q9	I am good at identifying what actions should be taken to minimize or alleviate something going wrong with a project. *Je suis compétent.e pour identifier les mesures à prendre pour minimiser ou atténuer les problèmes liés à un projet.*	A, B
Q10	I am good at recognizing that other team members' definition of what it means for something to "go wrong" may be different from my own. *Je suis compétent.e pour reconnaître que la définition des autres membres de l'équipe quant à ce que veut dire "tourner mal" peut différer de la mienne.*	B, D
Q11	When working on a project, I am good at asking myself if a project like this could have a positive impact on someone else's life. *Quand je travaille sur un projet, je suis compétent.e pour me demander si un projet comme celui-ci pourrait avoir un impact positif sur la vie de quelqu'un d'autre.*	C
Q12	When working on a project, I am good at asking myself if a project like this could have a negative impact on someone else's life. *Quand je travaille sur un projet, je suis compétent.e pour me demander si un projet comme celui-ci pourrait avoir un impact négatif sur la vie de quelqu'un d'autre.*	C
Q13	I am good at putting myself in the shoes of someone whose life could be affected by a project's results. *Je suis compétent.e pour me mettre à la place de quelqu'un dont la vie pourrait être affectée par les résultats d'un projet.*	C
Q14	I am good at identifying all the people who could be impacted by a project, no matter how directly or indirectly. *Je suis compétent.e pour identifier toutes les personnes qui pourraient être touchées par un projet, que ce soit directement ou indirectement.*	C
Q15	I am good at trying to understand the perspective of other team members. *Je suis compétent.e pour essayer de bien comprendre le point de vue des autres membres de l'équipe.*	D
Q16	I am good at making sure that all the necessary information is shared with other team members. *Je suis compétent.e pour m'assurer que toute l'information nécessaire est partagée avec les autres membres de l'équipe.*	D
Q17	I am good at explaining my ideas in ways that other people can understand. *Je suis compétent.e pour exposer mes idées de façon qu'elles soient bien comprises par les autres.*	D

Table B.1: (continued)

	Questions	Factors
Q18	When someone disagrees with me, I am good at paying close attention to see if I can learn something from their alternative perspective. *Quand quelqu'un est en désaccord avec moi, je suis compétent.e pour voir si je peux apprendre quelque-chose de leur point de vue.*	D
Q19	I can normally work productively with another team member even if I am angry or frustrated with them. *Je peux généralement travailler de façon productive avec un autre membre de l'équipe même si je suis en colère ou frustré contre lui.*	D
Q20	I am good at recognizing the knowledge and skills of different professions involved in a project team. *Je suis compétent.e pour reconnaître les connaissances et les compétences des différentes disciplines engagées dans un projet d'équipe.*	E
Q21	I am good at being sensitive to the way in which different professions may use the same word. *Je suis sensible à la façon dont différentes disciplines peuvent utiliser le même mot.*	E
Q22	I am good at clarifying with people from other professions how their knowledge and skills contribute to each stage of a project. *Je suis compétent.e pour clarifier avec les personnes d'autres disciplines comment leurs connaissances et leurs compétences contribuent à chaque étape d'un projet.*	E
Q23	I am good at identifying the skills or knowledge that other professions in the team have, which I should try to develop. *Je suis compétent.e pour identifier les connaissances ou compétences dont disposent les personnes des autres disciplines au sein de l'équipe et que je devrais développer moi-même.*	E
Q24	I am good at sharing responsibility with the other professions in the team for the overall success of a project. *Je suis compétent.e.e pour partager les responsabilités entre les personnes de différentes disciplines dans l'équipe afin d'assurer la réussite globale d'un projet.*	E

C

OMOD: Example Files

Content of the YAML example file given to students and that serves as a template to configure the optimizations:

```yaml
# Topology allows to test different number of gearpairs (here: 1 motor, 2 gearpairs)
topology:
  - motor
  - gearpair
  - gearpair
# Objectives to maximize (1) or minimize (-1)
# The order is kept for the graph of the results x -> 1, y -> 2, z (color) ->3
# Choose from the available variables
objectives:
  - ["cost", -1]
  - ["hmean_security_f", 1]
  - ["passthrough_axes", -1]
# Objectives to maximize (1) or minimize (-1)
# The order is kept for the graph of the results x -> 1, y -> 2, z (color) ->3
# Choose from the available variables (variables cannot appear twice)
constraints:
  - interference
  - contact_ratio
  - ss1
  - ss2
  - security_f
  - cycles_left_1
  - cycles_left_2
  - collision
  - i_tot
# Under constraints_specs you can specify the way and the value the form the constraint.
# In the example below, it means that i_tot is >= 1600

# The following constraints are taken into account the following way if you add them to the list above
# without the need to specify them here:
```

```yaml
# interference >= 0 <-- means no interference
# contact_ratio >= 1.1 <-- if you want to change this email me
# ss1 >= -5 <-- if you want to change this email me
# ss2 >= -5 <-- if you want to change this email me
# cycles_left_1 >= 0
# cycles_left_2 >= 0
# assembly_constraint <= 0 <-- meaning assembly is feasible if this is 0 (not to confuse
↪  with assembly score)
# passthrough_axes <= 0 <-- meaning all axes passthrough
# collision <= 0 <-- meaning no collision

constraints_specs:
  i_tot: [">", 1600]

# Optimization settings
# nfe = Number of function evaluations
# mu = Number of solutions evaluated per generation
optimization:
  nfe: 1000
  mu: 40

# Specify the bounds (lower / upper) allowed for the design
bounds:
  gear:
    allowed_materials: ['Ultramid-A3EG6', 'Genestar-N1001A', 'POM', 'Grivory-HT2V-5H',
    ↪  'Grivory-GV-4H']  # <-- set materials to consider
    Z1: [9, 40]
    x1: [-0.1, 0.6]
    Z2: [40, 100]
    x2: [-0.6, 0.6]
    m: [0.3, 1.0]
    b: [5., 15.]
    disp: [-40., 40.]
    angle: [-3.15, 3.15]
  motor:
    allowed_motors: ['A', 'B', 'C', 'D', 'E', 'F']  # <-- set motors to consider
    ff: [0.6, 0.8]
    r_scale: [0.5, 1.]

# Specify the operating points of the output to test the actuator
op:
  - speed: 0.314
    torque: 0.1
    temperature: 40
    ncycles: 10000
    type: "running"
    V: 12
    imax: 0.5
  - speed: 0.0056
    torque: 0.2
    temperature: -40
    ncycles: 10000
    type: "stall"
    V: 12
    imax: 0.5

# Settings of the model
settings:
  shaft_r: 1.25  # default radius to consider for shaft
  raddii:  # possibility to specify the radius of a particular shaft, here the 4th shaft
    3: 2
  limit_torque: True  # if True, motor torque is reduced if the output torque is higher than requested
  shaft_overlap: True  # if True allows multiple parts (pinion-wheel, pinion-wheel) to be on the same shaft

# For info only
variables:
  - cost  # sum of the cost of motor + gears
  - cost_with_hull  # cost + chull_area*1.5*rho*cost of Ultramid
  - min_torque_excess  # smallest torque excess (t_effective - t_desired) over all operating points
  - mean_torque_excess # mean torque excess over all operating points
  - passthrough_axes # 0 all axis passthrough / >= 0 some axis are "blocked" by other gears
  - assembly_constraint  # only constraint, 0 if assembly feasible, 1 if not feasible
  - assembly_score  # number of minimum asssembly steps (integer variable) / if greater than number of components --> assembly not feasible
  - hmean_security_f  # harmonic mean of the effective flexion security factor over all gear pairs
  - i_tot  # total transmission ratio from electric signal to output : i_tot = i_tot_gp * Nm
  - i_tot_gp  # total transmission ratio of gears
# Bounding box dimensions --> smallest axis aligned box that fits system (gears + motor only)
  - bb_dx
  - bb_dy
  - bb_dz  # z axis aligned with output shaft
# Oriented bouding box dimensions : rotation matrix optimized to have to smallest box that fits the system
# No alignement guarantee
  - obb_dx
  - obb_dy
  - obb_dz
# Minimum flexion security factor over all gear pairs shifted by -1 (security_f = 0.1 --> real security factor = 1.1)
  - security_f
  - cycles_left_1 # minimum number of cycles left over all pinions
  - cycles_left_2 # minimum number of cycles left over all wheels
  - ss1
  - ss2
  - interference
  - contact_ratio
  - collision  # 0 <-- no collision / 1 <-- every face is colliding
  - compactness  # volumes of components (gears + motor) / convex hull volume
  - bb_compactness  # volumes of components (gears + motor) / bounding box volume
  - bb_volume  # bounding box volume
  - obb_compactness # volumes of components (gears + motor) / oriented bounding box
   ↪  volume
  - obb_volume  # oriented bounding box volume
  - chull_volume  # convex hull volume
```

Content of the YAML example file to start full actuator evaluations:

```
topology:
  - motor
  - gearpair
  - gearpair
actuator:
  0_motor_name: 'B'
  0_fill_factor: 1.0
  0_r_scale: 1.0
  1_Z1: 13
  1_Z2: 67
  1_b: 10
  1_m: 0.5
  1_mat1: 'POM'
  1_mat2: 'POM'
  1_x1: 0.3
  1_x2: -0.3
  1_disp: 5
  1_angle: 0
  2_Z1: 17
  2_Z2: 70
  2_b: 10
  2_m: 0.5
  2_mat1: 'POM'
  2_mat2: 'POM'
  2_x1: 0.0
  2_x2: -0.0
  2_disp: 5
  2_angle: 1.2
settings:
  shaft_r: 1.25
  limit_torque: True
op:
  - speed: 0.314
    torque: 0.1
    temperature: 40
    ncycles: 80000
    type: "running"
    V: 12
    imax: 0.5
  - speed: 0.0056
    torque: 0.3
    temperature: -40
    ncycles: 80000
    type: "stall"
    V: 14
    imax: 0.5
```

Bibliography

[1] P. Acarnley. *Stepping Motors: A Guide to Theory and Practice*. IET, Jan. 2002. 174 pp.

[2] P. Aeby et al. “The Impact of Gender on Engineering Students’ Group Work Experiences”. In: *International Journal of Engineering Education* 35.3 (2019), pp. 756–765.

[3] S. Ahmed, K. M. Wallace, and L. T. Blessing. “Understanding the Differences between How Novice and Experienced Designers Approach Design Tasks”. In: *Research in Engineering Design* 14.1 (Feb. 1, 2003), pp. 1–11. DOI: 10.1007/s00163-002-0023-z.

[4] A. Alsin. *Inside Tesla – And What’s Really Disrupting The Automotive Industry*. Forbes. Sept. 14, 2017. URL: https://www.forbes.com/sites/aalsin/2017/09/14/inside-tesla-and-whats-really-disrupting-the-automotive-industry/ (visited on 10/21/2020).

[5] M. Asafuddoula et al. “An Adaptive Constraint Handling Approach Embedded MOEA/D”. In: *2012 IEEE Congress on Evolutionary Computation*. 2012 IEEE Congress on Evolutionary Computation. June 2012, pp. 1–8. DOI: 10.1109/CEC.2012.6252868.

[6] C. Audet and J. E. Dennis. “Mesh Adaptive Direct Search Algorithms for Constrained Optimization”. In: *SIAM Journal on Optimization* 17.1 (Jan. 1, 2006), pp. 188–217. DOI: 10.1137/040603371.

[7] J. Bader and E. Zitzler. “HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization”. In: *Evol. Comput.* 19.1 (Mar. 2011), pp. 45–76. DOI: 10.1162/EVCO_a_00009.

[8] B. J. Barron et al. “Doing With Understanding: Lessons From Research on Problem- and Project-Based Learning”. In: *Journal of the Learning Sciences* 7.3-4 (July 1998), pp. 271–311. DOI: 10.1080/10508406.1998.9672056.

[9] J. Blank and K. Deb. “Pymoo: Multi-Objective Optimization in Python”. In: *IEEE Access* 8 (2020), pp. 89497–89509. DOI: 10.1109/ACCESS.2020.2990567.

[10] L. T. Blessing and A. Chakrabarti. *DRM, a Design Research Methodology*. London: Springer London, 2009. DOI: 10.1007/978-1-84882-587-1.

[11] Bokeh Development Team. *Bokeh: Python Library for Interactive Visualization*. 2014.

[12] D. Brockhoff, T.-D. Tran, and N. Hansen. “Benchmarking Numerical Multiobjective Optimizers Revisited”. In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (Madrid, Spain). GECCO ’15. New York, NY, USA: ACM, 2015, pp. 639–646. DOI: 10.1145/2739480.2754777.

[13] A. F. Cabrera, C. L. Colbeck, and P. T. Terenzini. “Developing Performance Indicators for Assessing Classroom Teaching Practices and Student Learning”. In: *Research in Higher Education* 42.3 (June 2001), pp. 327–352. DOI: 10.1023/A:1018874023323.

[14] J. Cagan, K. Shimada, and S. Yin. “A Survey of Computational Approaches to Three-Dimensional Layout Problems”. In: *Computer-Aided Design* 34.8 (July 1, 2002), pp. 597–611. DOI: 10.1016/S0010-4485(01)00109-9.

[15] J. Cagan et al. “A Framework for Computational Design Synthesis: Model and Applications”. In: *Journal of Computing and Information Science in Engineering* 5.3 (Sept. 1, 2005), pp. 171–181. DOI: 10.1115/1.2013289.

[16] L. Carter. “Ideas for Adding Soft Skills Education to Service Learning and Capstone Courses for Computer Science Students”. In: *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. SIGCSE ’11. Dallas, TX, USA: Association for Computing Machinery, Mar. 2011, pp. 517–522. DOI: 10.1145/1953163.1953312.

[17] E. A. Cech. “Culture of Disengagement in Engineering Education?” In: *Science, Technology, & Human Values* 39.1 (Jan. 1, 2014), pp. 42–72. DOI: 10.1177/0162243913504305.

[18] A. Chakrabarti, S. Morgenstern, and H. Knaab. “Identification and Application of Requirements and Their Impact on the Design Process: A Protocol Study”. In: *Research in Engineering Design* 15.1 (Mar. 1, 2004), pp. 22–39. DOI: 10.1007/s00163-003-0033-5.

[19] A. Chakrabarti et al. “Computer-Based Design Synthesis Research: An Overview”. In: *Journal of Computing and Information Science in Engineering* 11.2 (June 1, 2011). DOI: 10.1115/1.3593409.

[20] J. Chen, A. Kolmos, and X. Du. “Forms of Implementation and Challenges of PBL in Engineering Education: A Review of Literature”. In: *European Journal of Engineering Education* 0.0 (Feb. 2020), pp. 1–26. DOI: 10.1080/03043797.2020.1718615.

[21] L. Chen et al. “An Artificial Intelligence Based Data-Driven Approach for Design Ideation”. In: *Journal of Visual Communication and Image Representation* 61 (May 1, 2019), pp. 10–22. DOI: 10.1016/j.jvcir.2019.02.009.

[22] W. Chen and F. Ahmed. “PaDGAN: Learning to Generate High-Quality Novel Designs”. In: *Journal of Mechanical Design* 143.031703 (Nov. 10, 2020). DOI: 10.1115/1.4048626.

[23] C. A. Coello Coello and M. S. Lechuga. “MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization”. In: *Proceedings of the Evolutionary Computation on 2002. CEC ’02. Proceedings of the 2002 Congress - Volume 02*. CEC ’02. USA: IEEE Computer Society, May 12, 2002, pp. 1051–1056.

[24] C. A. Coello Coello. “Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems”. In: *Computers in Industry* 41.2 (Mar. 1, 2000), pp. 113–127. DOI: 10.1016/S0166-3615(99)00046-9.

[25] C. A. Coello Coello and N. C. Cortés. “Solving Multiobjective Optimization Problems Using an Artificial Immune System”. In: *Genetic Programming and Evolvable Machines* 6.2 (June 1, 2005), pp. 163–190. DOI: 10.1007/s10710-005-6164-x.

[26] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2. ed. Genetic and Evolutionary Computation Series. New York, NY: Springer, 2007. 800 pp.

[27] A. R. Costa et al. “Impact of Interdisciplinary Learning on the Development of Engineering Students’ Skills”. In: *European Journal of Engineering Education* 44.4 (July 2019), pp. 589–601. DOI: 10.1080/03043797.2018.1523135.

[28] W. Cox and L. While. “Improving and Extending the HV4D Algorithm for Calculating Hypervolume Exactly”. In: *AI 2016: Advances in Artificial Intelligence*. Ed. by B. H. Kang and Q. Bai. Springer International Publishing, 2016, pp. 243–254.

[29] S. Craps et al. “Professional Roles and Employability of Future Engineers”. In: *Proceedings of the 45th SEFI Annual Conference 2017 - Education Excellence for Sustainability, SEFI 2017*. European Society for Engineering Education SEFI, Sept. 2017, pp. 499–507.

[30] E. F. Crawley et al. *Rethinking Engineering Education: The CDIO Approach*. 2nd ed. 2014. Cham: Springer International Publishing : Imprint: Springer, 2014. DOI: 10.1007/978-3-319-05561-9.

[31] M. L. Cruz, G. N. Saunders-Smits, and P. Groen. “Evaluation of Competency Methods in Engineering Education: A Systematic Review”. In: *European Journal of Engineering Education* 45.5 (Sept. 2020), pp. 729–757. DOI: 10.1080/03043797.2019.1671810.

[32] B. Danchilla. “Three.Js Framework”. In: *Beginning WebGL for HTML5*. Berkeley, CA: Apress, 2012, pp. 173–203. DOI: 10.1007/978-1-4302-3997-0_7.

[33] I. Das and J. Dennis. “Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems”. In: *SIAM Journal on Optimization* 8.3 (Aug. 1, 1998), pp. 631–657. DOI: 10.1137/S1052623496307510.

[34] Dawson-Haggerty et al. *Trimesh*. Version 3.2.0. Dec. 8, 2019.

[35] F.-M. De Rainville et al. “DEAP: A Python Framework for Evolutionary Algorithms”. In: *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*. GECCO ’12. New York, NY, USA: ACM, 2012, pp. 85–92. DOI: 10.1145/2330784.2330799.

[36] K. Deb and H. Jain. “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints”. In: *IEEE Transactions on Evolutionary Computation* 18.4 (Aug. 2014), pp. 577–601. DOI: 10.1109/TEVC.2013.2281535.

[37] K. Deb et al. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197. DOI: 10.1109/4235.996017.

[38] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, July 5, 2001. 540 pp.

[39] K. Deb and R. B. Agrawal. “Simulated Binary Crossover for Continuous Search Space”. In: *Complex systems* 9.2 (1995), pp. 115–148.

[40] K. Deb and S. Jain. “Multi-Speed Gearbox Design Using Multi-Objective Evolutionary Algorithms”. In: *Journal of Mechanical Design* 125.3 (2003), p. 609. DOI: 10.1115/1.1596242.

[41] K. Deb, A. Pratap, and T. Meyarivan. “Constrained Test Problems for Multi-Objective Evolutionary Optimization”. In: *Evolutionary Multi-Criterion Optimization*. Ed. by E. Zitzler et al. Springer Berlin Heidelberg, 2001, pp. 284–298.

[42] K. Deb et al. “Scalable Test Problems for Evolutionary Multiobjective Optimization”. In: *Evolutionary Multiobjective Optimization*. Ed. by A. Abraham, L. Jain, and R. Goldberg. Advanced Information and Knowledge Processing. Springer London, 2005, pp. 105–145. DOI: 10.1007/1-84628-137-7_6.

[43] M. Deininger et al. “Novice Designers’ Use of Prototypes in Engineering Design”. In: *Design Studies* 51 (July 2017), pp. 25–65. DOI: 10.1016/j.destud.2017.04.002.

[44] Design Council. *What Is the Framework for Innovation? Design Council’s Evolved Double Diamond*. Design Council. Mar. 17, 2015. URL: https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond (visited on 02/10/2021).

[45] F. Y. Edgeworth. *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. London, UK: P. Keagan, 1881.

[46] K. Edström and A. Kolmos. “PBL and CDIO: Complementary Models for Engineering Education Development”. In: *European Journal of Engineering Education* 39.5 (Sept. 2014), pp. 539–555. DOI: 10.1080/03043797.2014.895703.

[47] ENAEE. *EUR-ACE Label Authorisation Process*. ENAEE, Mar. 2017.

[48] L. J. Eshelman and J. D. Schaffer. “Real-Coded Genetic Algorithms and Interval-Schemata”. In: *Foundations of Genetic Algorithms*. Ed. by L. D. Whitley. Vol. 2. Foundations of Genetic Algorithms. Elsevier, Jan. 1, 1993, pp. 187–202. DOI: 10.1016/B978-0-08-094832-4.50018-0.

[49] Z. Fan et al. “An Improved Epsilon Constraint Handling Method Embedded in MOEA/D for Constrained Multi-Objective Optimization Problems”. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). Dec. 2016, pp. 1–8. DOI: 10.1109/SSCI.2016.7850224.

[50] Z. Fan et al. “An Improved Epsilon Constraint-Handling Method in MOEA/D for CMOPs with Large Infeasible Regions”. In: *Soft Computing* 23.23 (Dec. 1, 2019), pp. 12491–12510. DOI: 10.1007/s00500-019-03794-x.

[51] Z. Fan et al. “Difficulty Adjustable and Scalable Constrained Multiobjective Test Problem Toolkit”. In: *Evolutionary Computation* (May 23, 2019), pp. 1–40. DOI: 10.1162/evco_a_00259.

[52] Z. Fan et al. “MOEA/D with Angle-Based Constrained Dominance Principle for Constrained Multi-Objective Optimization Problems”. In: *Applied Soft Computing* 74 (Jan. 1, 2019), pp. 621–633. DOI: 10.1016/j.asoc.2018.10.027.

[53] Z. Fan et al. “Push and Pull Search for Solving Constrained Multi-Objective Optimization Problems”. In: *Swarm and Evolutionary Computation* 44 (Feb. 1, 2019), pp. 665–679. DOI: 10.1016/j.swevo.2018.08.017.

[54] M. Farina and P. Amato. “On the Optimal Solution Definition for Many-Criteria Optimization Problems”. In: *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622).* 2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622). 2002, pp. 233–238. DOI: 10.1109/NAFIPS.2002.1018061.

[55] J. R. Flynn. *What Is Intelligence ? : Beyond the Flynn Effect*. 1st expanded paperback ed. [with 3 new chapters]. Cambridge ; New York: Cambridge Univ. Press, 2009. 259 p.

[56] F.-A. Fortin et al. “DEAP: Evolutionary Algorithms Made Easy”. In: *Journal of Machine Learning Research* 13 (Jul 2012), pp. 2171–2175.

[57] C. B. Frey and M. A. Osborne. “The Future of Employment: How Susceptible Are Jobs to Computerisation?” In: *Technological Forecasting and Social Change* 114 (Jan. 1, 2017), pp. 254–280. DOI: 10.1016/j.techfore.2016.08.019.

[58] H. Fukumoto and A. Oyama. “A Generic Framework for Incorporating Constraint Handling Techniques into Multi-Objective Evolutionary Algorithms”. In: *Applications of Evolutionary Computation*. International Conference on the Applications of Evolutionary Computation. Springer, Cham, Apr. 3, 2018, pp. 634–649. DOI: 10.1007/978-3-319-77538-8_43.

[59] H. Fukumoto and A. Oyama. “Benchmarking Multiobjective Evolutionary Algorithms and Constraint Handling Techniques on a Real-World Car Structure Design Optimization Benchmark Problem”. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO ’18. New York, NY, USA: ACM, 2018, pp. 177–178. DOI: 10.1145/3205651.3205754.

[60] R. d. P. Garcia et al. “A Rank-Based Constraint Handling Technique for Engineering Design Optimization Problems Solved by Genetic Algorithms”. In: *Computers & Structures* 187 (July 15, 2017), pp. 77–87. DOI: 10.1016/j.compstruc.2017.03.023.

[61] H. Geng et al. “Infeasible Elitists and Stochastic Ranking Selection in Constrained Evolutionary Multi-Objective Optimization”. In: *Simulated Evolution and Learning*. Ed. by T.-D. Wang et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 336–344. DOI: 10.1007/11903697_43.

[62] J. S. Gero and U. Kannengiesser. “The Situated Function–Behaviour–Structure Framework”. In: *Design Studies* 25.4 (July 1, 2004), pp. 373–391. DOI: 10.1016/j.destud.2003.10.010.

[63] D. Gijbels et al. “Effects of Problem-Based Learning: A Meta-Analysis From the Angle of Assessment”. In: *Review of Educational Research* 75.1 (Mar. 2005), pp. 27–61. DOI: 10.3102/00346543075001027.

[64] P. M. Grignon and G. M. Fadel. “A GA Based Configuration Design Optimization Method”. In: *Journal of Mechanical Design* 126.1 (Mar. 11, 2004), pp. 6–15. DOI: 10.1115/1.1637656.

[65] M. Grimheden. “From Capstone Courses To Cornerstone Projects: Transferring Experiences From Design Engineering Final Year Students To First Year Students”. In: *114th Annual ASEE Conference and Exposition, 2007*. ASEE Conferences, June 2007, pp. 12.768.1–12.768.12.

[66] S. Grover and R. D. Pea. “Computational Thinking: A Competency Whose Time Has Come”. In: *Computer Science Education: Perspectives on Teaching and Learning in School*. London ; New York: Bloomsbury Academic, 2018.

[67] J. Gu et al. “Parameter Optimization Design of Two-Planetary-Gear Power-Split Hybrid System Configuration Using the RAD-MOPSO Algorithm”. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* (July 24, 2020), p. 0954407020937527. DOI: 10.1177/0954407020937527.

[68] D. Hadka and P. Reed. “Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework”. In: *Evolutionary Computation* 21.2 (May 2013), pp. 231–259. DOI: 10.1162/EVCO_a_00075.

[69] J. Han et al. “A Computational Tool for Creative Idea Generation Based on Analogical Reasoning and Ontology”. In: *AI EDAM* 32.4 (Nov. 2018), pp. 462–477. DOI: 10.1017/S0890060418000082.

[70] J. Hattie. *Visible Learning: A Synthesis of over 800 Meta-Analyses Relating to Achievement*. London ; New York: Routledge, 2009.

[71] R. Hernandez-Linares et al. “Transversal Competences of University Students of Engineering”. In: *Croatian Journal of Education : Hrvatski časopis za odgoj i obrazovanje* 17.2 (June 2015), pp. 383–409. DOI: 10.15516/cje.v17i2.1062.

[72] C. E. Hmelo, D. L. Holton, and J. L. Kolodner. “Designing to Learn About Complex Systems”. In: *Journal of the Learning Sciences* 9.3 (July 1, 2000), pp. 247–298. DOI: 10.1207/S15327809JLS0903_2.

[73] S. Holm. “A Simple Sequentially Rejective Multiple Test Procedure”. In: *Scandinavian Journal of Statistics* 6.2 (1979), pp. 65–70.

[74] N. Houssami et al. “Artificial Intelligence for Breast Cancer Screening: Opportunity or Hype?” In: *The Breast* 36 (Dec. 1, 2017), pp. 31–33. DOI: 10.1016/j.breast.2017.09.003.

[75] T. Howard, S. Culley, and E. Dekoninck. “Describing the Creative Design Process by the Integration of Engineering Design and Cognitive Psychology Literature”. In: *Design Studies* 29.2 (Mar. 2008), pp. 160–180. DOI: 10.1016/j.destud.2008.01.001.

[76] S. Howe. “Where Are We Now? Statistics on Capstone Courses Nationwide”. In: *Advances in Engineering Education* 2.1 (2010), p. 27.

[77] S. Howe, L. Rosenbauer, and S. Poulos. “The 2015 Capstone Design Survey Results: Current Practices and Changes over Time”. In: *International Journal of Engineering Education* 33.5 (2017), pp. 393–1421.

[78] S. Huband et al. “A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit”. In: *IEEE Transactions on Evolutionary Computation* 10.5 (Oct. 2006), pp. 477–506. DOI: 10.1109/TEVC.2005.861417.

[79] International Organization for Standardization. *Calculation of Load Capacity of Spur and Helical Gears*. ISO 6336:2006. Geneva, Switzerland: ISO, 2006.

[80] International Organization for Standardization. *Cylindrical Gears for General and Heavy Engineering – Standard Basic Rack Tooth Profile*. ISO 53:1998. Geneva, Switzerland: ISO, 1998.

[81] International Organization for Standardization. *Gears – Cylindrical Involute Gears and Gear Pairs – Concepts and Geometry*. ISO 21771:2007. Geneva, Switzerland: ISO, 2007.

[82] International Organization for Standardization. *Vocabulary of Gear Terms – Part 1: Definitions Related to Geometry*. ISO 1122-1:1998. Geneva, Switzerland: ISO, 1998.

[83] S. Isaac and R. Tormey. “Undergraduate Group Projects: Challenges and Learning Experiences”. In: *Engineering Leaders Conference 2014 on Engineering Education*. Vol. 2015. Hamad bin Khalifa University Press (HBKU Press), Aug. 2015, p. 19. DOI: 10.5339/qproc.2015.elc2014.19.

[84] H. Ishibuchi, N. Akedo, and Y. Nojima. “Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems”. In: *IEEE Transactions on Evolutionary Computation* 19.2 (Apr. 2015), pp. 264–283. DOI: 10.1109/TEVC.2014.2315442.

[85] H. Ishibuchi et al. “Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes”. In: *IEEE Transactions on Evolutionary Computation* 21.2 (Apr. 2017), pp. 169–190. DOI: 10.1109/TEVC.2016.2587749.

[86] Itani Mona and Srour Issam. “Engineering Students’ Perceptions of Soft Skills, Industry Expectations, and Career Aspirations”. In: *Journal of Professional Issues in Engineering Education and Practice* 142.1 (Jan. 2016), p. 04015005. DOI: 10.1061/(ASCE)EI.1943-5541.0000247.

[87] H. Jain and K. Deb. “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach”. In: *IEEE Transactions on Evolutionary Computation* 18.4 (Aug. 2014), pp. 602–622. DOI: 10.1109/TEVC.2013.2281534.

[88] M. S. Jain and T. F. Massoud. “Predicting Tumour Mutational Burden from Histopathological Images Using Multiscale Deep Learning”. In: *Nature Machine Intelligence* 2.6 (6 June 2020), pp. 356–362. DOI: 10.1038/s42256-020-0190-5.

[89] M. A. Jan and Q. Zhang. “MOEA/D for Constrained Multiobjective Optimization: Some Preliminary Experimental Results”. In: 2010 UK Workshop on Computational Intelligence (UKCI). Sept. 2010, pp. 1–6. DOI: 10.1109/UKCI.2010.5625585.

[90] M. B. Jensen, C. W. Elverum, and M. Steinert. “Eliciting Unknown Unknowns with Prototypes: Introducing Prototrials and Prototrial-Driven Cultures”. In: *Design Studies* 49 (Mar. 2017), pp. 1–31. DOI: 10.1016/j.destud.2016.12.002.

[91] JPNSEC. *3rd Evolutionary Computation Competition*. 2019.

[92] U. Kannengiesser and J. S. Gero. “Can Pahl and Beitz’ Systematic Approach Be a Predictive Model of Designing?” In: *Design Science* 3 (2017). DOI: 10.1017/dsj.2017.24.

[93] R. G. Klaassen. “Interdisciplinary Education: A Case Study”. In: *European Journal of Engineering Education* 43.6 (Nov. 2018), pp. 842–859. DOI: 10.1080/03043797.2018.1442417.

[94] J. Knowles, L. Thiele, and E. Zitzler. *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. 214. Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, Feb. 2006.

[95] A. Kolmos, E. de Graaff, and X. Du. “Diversity of PBL: - PBL Learning Principles and Models”. In: *Research on PBL Practice in Engineering Education* (2009), pp. 9–21.

[96] A. Kolmos and J. E. Holgaard. “Employability in Engineering Education: Are Engineering Students Ready for Work?” In: *The Engineering-Business Nexus: Symbiosis, Tension and Co-Evolution*. Ed. by S. H. Christensen et al. Philosophy of Engineering and Technology. Cham: Springer International Publishing, 2019, pp. 499–520. DOI: 10.1007/978-3-319-99636-3_22.

[97] C. Königseder and K. Shea. “Comparing Strategies for Topologic and Parametric Rule Application in Automated Computational Design Synthesis”. In: *Journal of Mechanical Design* 138.1 (Jan. 1, 2016), p. 011102. DOI: 10.1115/1.4031714.

[98] W. H. Kruskal and W. A. Wallis. “Use of Ranks in One-Criterion Variance Analysis”. In: *Journal of the American Statistical Association* 47.260 (Dec. 1, 1952), pp. 583–621. DOI: 10.1080/01621459.1952.10483441.

[99] M. Laperrouza and R. Tormey. “Developing a Questionnaire to Assess Interdisciplinary Project Management Skills”. In: *Lunch&LEARN*. EPFL, Lausanne, Switzerland, Feb. 2019.

[100] L. R. Lattuca et al. “Supporting the Development of Engineers’ Interdisciplinary Competence”. In: *Journal of Engineering Education* 106.1 (2017), pp. 71–97. DOI: 10.1002/jee.20155.

[101] S. Le Digabel. “Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm”. In: *ACM Transactions on Mathematical Software* 37.4 (Feb. 1, 2011), 44:1–44:15. DOI: 10.1145/1916461.1916468.

[102] E. S. Lemaitre. “Actuator Synthesis and Optimization: Evaluation of Graph-Based Approaches”. Master thesis. Lausanne, Switzerland: EPFL, June 19, 2020.

[103] H. Li and Q. Zhang. “Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 13.2 (Apr. 2009), pp. 284–302. DOI: 10.1109/TEVC.2008.925798.

[104] J. Li et al. “A Comparative Study of Constraint-Handling Techniques in Evolutionary Constrained Multiobjective Optimization”. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*. 2016 IEEE Congress on Evolutionary Computation (CEC). July 2016, pp. 4175–4182. DOI: 10.1109/CEC.2016.7744320.

[105] K. Li et al. “Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization”. In: *IEEE Transactions on Evolutionary Computation* 23.2 (Apr. 2019), pp. 303–315. DOI: 10.1109/TEVC.2018.2855411.

[106] Y.-s. Lin et al. “A Method and Software Tool for Automated Gearbox Synthesis”. In: *Volume 5: 35th Design Automation Conference, Parts A and B*. ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. San Diego, California, USA: ASME, 2009, pp. 111–121. DOI: 10.1115/DETC2009-86935.

[107] Z.-Z. Liu and Y. Wang. “Handling Constrained Multiobjective Optimization Problems With Constraints in Both the Decision and Objective Spaces”. In: *IEEE Transactions on Evolutionary Computation* 23.5 (Oct. 2019), pp. 870–884. DOI: 10.1109/TEVC.2019.2894743.

[108] M. Lopez-Ibanez and T. Stutzle. “The Automatic Design of Multiobjective Ant Colony Optimization Algorithms”. In: *IEEE Transactions on Evolutionary Computation* 16.6 (Dec. 2012), pp. 861–875. DOI: 10.1109/TEVC.2011.2182651.

[109] M. López-Ibáñez and T. Stützle. “Automatic Configuration of Multi-Objective ACO Algorithms”. In: *Swarm Intelligence*. International Conference on Swarm Intelligence. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sept. 8, 2010, pp. 95–106. DOI: 10.1007/978-3-642-15461-4_9.

[110] M. López-Ibáñez et al. “The Irace Package: Iterated Racing for Automatic Algorithm Configuration”. In: *Operations Research Perspectives* 3 (Jan. 1, 2016), pp. 43–58. DOI: 10.1016/j.orp.2016.09.002.

[111] A. F. Lourenço. “Testing of Low-Loss Polymer Gears”. Master thesis. Porto, Portugal: Faculdade de Engenharia da Universidade do Porto, July 20, 2015.

[112] Z. Ma and Y. Wang. “Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons”. In: *IEEE Transactions on Evolutionary Computation* 23.6 (Dec. 2019), pp. 972–986. DOI: 10.1109/TEVC.2019.2896967.

[113] K. M. Malan and A. P. Engelbrecht. “A Progressive Random Walk Algorithm for Sampling Continuous Fitness Landscapes”. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. 2014 IEEE Congress on Evolutionary Computation (CEC). July 2014, pp. 2507–2514. DOI: 10.1109/CEC.2014.6900576.

[114] K. M. Malan and I. Moser. “Constraint Handling Guided by Landscape Analysis in Combinatorial and Continuous Search Spaces”. In: *Evolutionary Computation* 27.2 (Mar. 12, 2018), pp. 267–289. DOI: 10.1162/evco_a_00222.

[115] K. M. Malan, J. F. Oberholzer, and A. P. Engelbrecht. “Characterising Constrained Continuous Optimisation Problems”. In: 2015 IEEE Congress on Evolutionary Computation (CEC). Sendai, Japan: IEEE, May 2015, pp. 1351–1358. DOI: 10.1109/CEC.2015.7257045.

[116] R. Mallipeddi and P. N. Suganthan. “Ensemble of Constraint Handling Techniques”. In: *IEEE Transactions on Evolutionary Computation* 14.4 (Aug. 2010), pp. 561–579. DOI: 10.1109/TEVC.2009.2033582.

[117] R. Mallipeddi and P. N. Suganthan. “Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization”. In: *Nanyang Technological University, Singapore* (2010).

[118] S. Maneewongvatana and D. M. Mount. “Analysis of Approximate Nearest Neighbor Searching with Clustered Point Sets”. In: (Jan. 26, 1999).

[119] H. B. Mann and D. R. Whitney. “On a Test of Whether One of Two Random Variables Is Stochastically Larger than the Other”. In: *Annals of Mathematical Statistics* 18.1 (Mar. 1947), pp. 50–60. DOI: 10.1214/aoms/1177730491.

[120] J. McCarthy. “What Is Artificial Intelligence?” In: (2007), p. 15.

[121] J. McGourty. “Using Multisource Feedback in the Classroom: A Computer-Based Approach”. In: *IEEE Transactions on Education* 43.2 (May 2000), pp. 120–124. DOI: 10.1109/13.848062.

[122] E. Mezura-Montes and C. A. Coello Coello. “Constraint-Handling in Nature-Inspired Numerical Optimization: Past, Present and Future”. In: *Swarm and Evolutionary Computation* 1.4 (Dec. 1, 2011), pp. 173–194. DOI: 10.1016/j.swevo.2011.10.001.

[123] A. Mohan et al. “Professional Skills in the Engineering Curriculum”. In: *IEEE Transactions on Education* 53.4 (Nov. 2010), pp. 562–571. DOI: 10.1109/TE.2009.2033041.

[124] V. Mounier, C. Picard, and J. Schiffmann. “Data-Driven Pre-Design Tool for Small Scale Centrifugal Compressors in Refrigeration”. In: *Journal of Engineering for Gas Turbines and Power* (July 27, 2018). DOI: 10.1115/1.4040845.

[125] D. Muhammad and M. Asaduzzaman. “Friction and Wear of Polymer and Composites”. In: *Composites and Their Properties*. Ed. by N. Hu. InTech, Aug. 22, 2012. DOI: 10.5772/48246.

[126] “New Perspectives on Design Automation: Celebrating the 40th Anniversary of the ASME Design Automation Conference”. In: *Journal of Mechanical Design* 137.050301 (May 1, 2015). DOI: 10.1115/1.4030256.

[127] M. Newman. *Problem Based Learning: An Exploration of the Method and Evaluation of Its Effectiveness in a Continuing Nursing Education Programme.* Project on the effectiveness of problem based learning (PEPBL) research report. Middlesex: Middlesex University, 2004.

[128] G. Niemann and H. Winter. *Getriebe allgemein, Zahnradgetriebe - Grundlagen, Stirnradgetriebe*. 2., völlig neubearb. Aufl., 2. berichtigter Nachdr., korrigierter Nachdr. Maschinenelemente G. Niemann; H. Winter ; Bd. 2. Berlin: Springer, 2003. 376 pp.

[129] A. K. Noor. “AI and the Future of the Machine Design: Artificially Intelligent Systems Are Learning How to Develop New Products and Design. What Does That Leave the Engineers to Do?” In: *Mechanical Engineering* 139.10 (Oct. 1, 2017), pp. 38–43. DOI: 10.1115/1.2017-Oct-2.

[130] E. Nuhfer et al. “How Random Noise and a Graphical Convention Subverted Behavioral Scientists’ Explanations of Self-Assessment Data: Numeracy Underlies Better Alternatives”. In: *Numeracy* 10.1 (Jan. 2017). DOI: http://dx.doi.org/10.5038/1936-4660.10.1.4.

[131] E. Nuhfer et al. “Random Number Simulations Reveal How Random Noise Affects the Measurements and Graphical Portrayals of Self-Assessed Competency”. In: *Numeracy* 9.1 (Jan. 2016). DOI: http://dx.doi.org/10.5038/1936-4660.9.1.4.

[132] B. Oakley et al. “Turning Student Groups into Effective Teams”. In: *Journal of student centered learning* 2.1 (2004), pp. 9–34.

[133] A. Osyczka and S. Kundu. “A New Method to Solve Generalized Multicriteria Optimization Problems Using the Simple Genetic Algorithm”. In: *Structural optimization* 10.2 (Oct. 1, 1995), pp. 94–99. DOI: 10.1007/BF01743536.

[134] G. Pahl et al. *Engineering Design: A Systematic Approach*. 3rd ed. London: Springer-Verlag, 2007.

[135] J. Pan, S. Chitta, and D. Manocha. "FCL: A General Purpose Library for Collision and Proximity Queries". In: 2012 IEEE International Conference on Robotics and Automation (ICRA). St Paul, MN, USA: IEEE, May 2012, pp. 3859–3866. DOI: 10.1109/ICRA.2012.6225337.

[136] E. Panadero, J. Klug, and S. Järvelä. "Third Wave of Measurement in the Self-Regulated Learning Field: When Measurement and Intervention Come Hand in Hand". In: *Scandinavian Journal of Educational Research* 60.6 (Nov. 2016), pp. 723–735. DOI: 10.1080/00313831.2015.1066436.

[137] P. Y. Papalambros and D. J. Wilde. *Principles of Optimal Design: Modeling and Computation*. 3rd ed. Cambridge University Press, Jan. 9, 2017. DOI: 10.1017/9781316451038.

[138] V. Pareto. *Course of Political Economy*. Lausanne: F. Rouge, 1896.

[139] D. L. Paulhus and S. Vazire. "The Self-Report Method". In: *Handbook of Research Methods in Personality Psychology*. New York, NY, US: The Guilford Press, 2007, pp. 224–239.

[140] C. Picard and J. Schiffmann. "Realistic Constrained Multiobjective Optimization Benchmark Problems From Design". In: *IEEE Transactions on Evolutionary Computation* 25.2 (Apr. 2021), pp. 234–246. DOI: 10.1109/TEVC.2020.3020046.

[141] C. Picard and J. Schiffmann. "Automated Design Tool for Automotive Control Actuators". In: *IDETC-CIE2020*. ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 11B: 46th Design Automation Conference (DAC), Aug. 17, 2020. DOI: 10.1115/DETC2020-22390.

[142] C. Picard and J. Schiffmann. "Impacts of Constraints and Constraint Handling Strategies for Multi-Objective Mechanical Design Problems". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '18. New York, NY, USA: ACM, 2018, pp. 1341–1347. DOI: 10.1145/3205455.3205526.

[143] C. Picard and J. Schiffmann. *Multi-Objective Design of Actuators: Pareto Fronts*. Zenodo, May 13, 2020. DOI: 10.5281/zenodo.3824302.

[144] A. Planas-Lladó et al. "An Analysis of Teamwork Based on Self and Peer Evaluation in Higher Education". In: *Assessment & Evaluation in Higher Education* 0.0 (May 2020), pp. 1–17. DOI: 10.1080/02602938.2020.1763254.

[145] M. J. Prince and R. M. Felder. "Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases". In: *Journal of Engineering Education* 95.2 (Apr. 2006), pp. 123–138. DOI: 10.1002/j.2168-9830.2006.tb00884.x.

[146] Python Software Foundation. *Python Language Referene*. Version 3.7.

[147] E. Ramadi, S. Ramadi, and K. Nasr. “Engineering Graduates’ Skill Sets in the MENA Region: A Gap Analysis of Industry Expectations and Satisfaction”. In: *European Journal of Engineering Education* 41.1 (Jan. 2016), pp. 34–52. DOI: 10.1080/03043797.2015.1012707.

[148] T. Ray, K. Tai, and K. C. Seow. “Multiobjective Design Optimization by an Evolutionary Algorithm”. In: *Engineering Optimization* 33.4 (Apr. 1, 2001), pp. 399–424. DOI: 10.1080/03052150108940926.

[149] T. Ray et al. “Infeasibility Driven Evolutionary Algorithm for Constrained Optimization”. In: *Constraint-Handling in Evolutionary Optimization*. Ed. by E. Mezura-Montes. Studies in Computational Intelligence. Berlin, Heidelberg: Springer, 2009, pp. 145–165. DOI: 10.1007/978-3-642-00619-7_7.

[150] J. Régnier et al. “Optimal Design of Electrical Engineering Systems Using Pareto Genetic Algorithms”. In: *10th European Conference on Power Electronics and Applications, Toulouse*. Vol. 21. 2003, pp. 259–261.

[151] G. Reynders et al. “Rubrics to Assess Critical Thinking and Information Processing in Undergraduate STEM Courses”. In: *International Journal of STEM Education* 7.1 (Dec. 2020), p. 9. DOI: 10.1186/s40594-020-00208-5.

[152] J. T. Richardson et al. “Some Guidelines for Genetic Algorithms with Penalty Functions”. In: *Proceedings of the Third International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 191–197.

[153] E. Rigger, K. Shea, and T. Stankovic. “Task Categorisation for Identification of Design Automation Opportunities”. In: *Journal of Engineering Design* 29.3 (Mar. 4, 2018), pp. 131–159. DOI: 10.1080/09544828.2018.1448927.

[154] E. Rigger et al. “A Top-down Method for the Derivation of Metrics for the Assessment of Design Automation Potential”. In: *Journal of Engineering Design* 31.2 (Feb. 1, 2020), pp. 69–99. DOI: 10.1080/09544828.2019.1670786.

[155] T. Runarsson and X. Yao. “Stochastic Ranking for Constrained Evolutionary Optimization”. In: *IEEE Transactions on Evolutionary Computation* 4.3 (Sept. 2000), pp. 284–294. DOI: 10.1109/4235.873238.

[156] S. Salcedo-Sanz. “A Survey of Repair Methods Used as Constraint Handling Techniques in Evolutionary Algorithms”. In: *Computer Science Review* 3.3 (Aug. 1, 2009), pp. 175–192. DOI: 10.1016/j.cosrev.2009.07.001.

[157] S. Santoso. *Standard Handbook for Electrical Engineers, Seventeenth Edition*. 17th edition. New York, NY: McGraw-Hill Education, 2017.

[158] J. Savery. “Overview of Problem-Based Learning: Definitions and Distinctions”. In: *Interdisciplinary Journal of Problem-Based Learning* 1.1 (May 2006). DOI: 10.7771/1541-5015.1002.

[159] J. Schiffmann. “Integrated Design and Multi-Objective Optimization of a Single Stage Heat-Pump Turbocompressor”. In: *Journal of Turbomachinery* 137.7 (July 1, 2015), pp. 071002-071002–9. DOI: 10.1115/1.4029123.

[160] J. Schijve. “Fatigue of Structures and Materials in the 20th Century and the State of the Art”. In: *International Journal of Fatigue* 25.8 (Aug. 1, 2003), pp. 679–702. DOI: 10.1016/S0142-1123(03)00051-3.

[161] B. Schmitz and F. Perels. “Self-Monitoring of Self-Regulation during Math Homework Behaviour Using Standardized Diaries”. In: *Metacognition and Learning* 6.3 (Dec. 2011), pp. 255–273. DOI: 10.1007/s11409-011-9076-6.

[162] L. J. Shuman, M. Besterfield-Sacre, and J. McGourty. “The ABET “Professional Skills” — Can They Be Taught? Can They Be Assessed?” In: *Journal of Engineering Education* 94.1 (2005), pp. 41–55. DOI: 10.1002/j.2168-9830.2005.tb00828.x.

[163] D. Silver et al. “Mastering the Game of Go with Deep Neural Networks and Tree Search”. In: *Nature* 529.7587 (Jan. 28, 2016), pp. 484–489. DOI: 10.1038/nature16961.

[164] D. Silver et al. “Mastering the Game of Go without Human Knowledge”. In: *Nature* 550.7676 (7676 Oct. 2017), pp. 354–359. DOI: 10.1038/nature24270.

[165] G. Spaeth. “Automated Design Tools for Electro-Mechanical Actuators”. Master thesis. Lausanne: EPFL, Jan. 17, 2020.

[166] N. Srinivas and K. Deb. “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms”. In: *Evolutionary Computation* 2.3 (Sept. 1, 1994), pp. 221–248. DOI: 10.1162/evco.1994.2.3.221.

[167] V. Stanovov, S. Akhmedova, and E. Semenkin. “Combined Fitness–Violation Epsilon Constraint Handling for Differential Evolution”. In: *Soft Computing* (Mar. 10, 2020). DOI: 10.1007/s00500-020-04835-6.

[168] R. J. Sternberg. “The Concept of Intelligence and Its Role in Lifelong Learning and Success”. In: *American Psychologist* 52.10 (1997), pp. 1030–1037. DOI: 10.1037/0003-066X.52.10.1030.

[169] A. Swantner and M. I. Campbell. “Topological and Parametric Optimization of Gear Trains”. In: *Engineering Optimization* 44.11 (Nov. 1, 2012), pp. 1351–1368. DOI: 10.1080/0305215X.2011.646264.

[170] S. Szykman and J. Cagan. “Constrained Three-Dimensional Component Layout Using Simulated Annealing”. In: *Journal of Mechanical Design* 119.1 (Mar. 1, 1997), pp. 28–35. DOI: 10.1115/1.2828785.

[171] T. Takahama and S. Sakai. “Constrained Optimization by the ε Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites”. In: 2006 IEEE International Conference on Evolutionary Computation. July 2006, pp. 1–8. DOI: 10.1109/CEC.2006.1688283.

[172] T. Takahama and S. Sakai. “Efficient Constrained Optimization by the ε Constrained Differential Evolution with Rough Approximation”. In: *Evolutionary Constrained Optimization*. Ed. by R. Datta and K. Deb. Infosys Science Foundation Series. New Delhi: Springer India, 2015, pp. 157–180. DOI: 10.1007/978-81-322-2184-5_6.

[173] R. Tanabe and A. Oyama. “A Note on Constrained Multi-Objective Optimization Benchmark Problems”. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. 2017 IEEE Congress on Evolutionary Computation (CEC). June 2017, pp. 1127–1134. DOI: 10.1109/CEC.2017.7969433.

[174] R. Tanabe and H. Ishibuchi. “An Easy-to-Use Real-World Multi-Objective Optimization Problem Suite”. In: *Applied Soft Computing* 89 (Apr. 1, 2020), p. 106078. DOI: 10.1016/j.asoc.2020.106078.

[175] M. Tanaka et al. “GA-Based Decision Support System for Multicriteria Optimization”. In: 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century. Vol. 2. Oct. 1995, 1556–1561 vol.2. DOI: 10.1109/ICSMC.1995.537993.

[176] Y. Tian et al. “PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]”. In: *IEEE Computational Intelligence Magazine* 12.4 (Nov. 2017), pp. 73–87. DOI: 10.1109/MCI.2017.2742868.

[177] Y. Tolkach et al. “High-Accuracy Prostate Cancer Pathology Using Deep Learning”. In: *Nature Machine Intelligence* 2.7 (7 July 2020), pp. 411–418. DOI: 10.1038/s42256-020-0200-7.

[178] R. Tormey et al. “The Formal and Hidden Curricula of Ethics in Engineering Education”. In: *43rd Annual SEFI Conference*. 43rd Annual SEFI Conference. CONF. 2015.

[179] M. F. Torres, A. J. Sousa, and R. T. Torres. “Pedagogical and Technological Replanning: A Successful Case Study on Integration and Transversal Skills for Engineering Freshmen”. In: *International Journal of Technology and Design Education* 28.2 (June 2018), pp. 573–591. DOI: 10.1007/s10798-017-9399-y.

[180] TU Berlin. *Forests in Distress*. Sept. 29, 2020. URL: https://www.tu.berlin/en/about/profile/pressemitteilungen-nachrichten/2020/september/forests-in-distress/ (visited on 10/21/2020).

[181] D. G. Ullman. *The Mechanical Design Process*. McGraw-Hill Higher Education US, 1992.

[182] R. Vallat. “Pingouin: Statistics in Python”. In: *Journal of Open Source Software* 3.31 (Nov. 19, 2018), p. 1026. DOI: 10.21105/joss.01026.

[183] VDI-Fachbereich Getriebe und Maschinenelemente. *VDI 2736 Blatt 2 - Thermoplastic gear wheels - Cylindrical gears - Calculation of the load-carrying capacity*. Düsseldorf: VDI-Verlag GmbH, June 2014.

[184] VDI-Fachbereich Getriebe und Maschinenelemente. *VDI 2736 Blatt 3 - Thermoplastic gear wheels - Crossed helical gears - Mating cylindrical worm with helical gear - Calculation of the load-carrying capacity*. Düsseldorf: VDI-Verlag GmbH, May 2014.

[185] T. Voß, N. Hansen, and C. Igel. "Improved Step Size Adaptation for the MO-CMA-ES". In: ACM Press, 2010, p. 487. DOI: 10.1145/1830483.1830573.

[186] J. A. Vrugt and B. A. Robinson. "Improved Evolutionary Optimization from Genetically Adaptive Multimethod Search". In: *Proceedings of the National Academy of Sciences* 104.3 (Jan. 16, 2007), pp. 708–711. DOI: 10.1073/pnas.0610471104.

[187] I. Wald et al. "Embree: A Kernel Framework for Efficient CPU Ray Tracing". In: *ACM Transactions on Graphics* 33.4 (July 27, 2014), pp. 1–8. DOI: 10.1145/2601097.2601199.

[188] L. Wang et al. "Examination of Multi-Objective Optimization Method for Global Search Using DIRECT and GA". In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence).* 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). June 2008, pp. 2446–2451. DOI: 10.1109/CEC.2008.4631125.

[189] L. While, L. Bradstreet, and L. Barone. "A Fast Way of Calculating Exact Hypervolumes". In: *IEEE Transactions on Evolutionary Computation* 16.1 (Feb. 2012), pp. 86–95. DOI: 10.1109/TEVC.2010.2077298.

[190] O. Willner, J. Gosling, and P. Schönsleben. "Establishing a Maturity Model for Design Automation in Sales-Delivery Processes of ETO Products". In: *Computers in Industry* 82 (Oct. 1, 2016), pp. 57–68. DOI: 10.1016/j.compind.2016.05.003.

[191] C. Winberg et al. "Developing Employability in Engineering Education: A Systematic Review of the Literature". In: *European Journal of Engineering Education* 45.2 (Mar. 2020), pp. 165–180. DOI: 10.1080/03043797.2018.1534086.

[192] J. M. Wing. "Computational Thinking". In: *Communications of the ACM* 49.3 (Mar. 1, 2006), pp. 33–35. DOI: 10.1145/1118178.1118215.

[193] J. C. F. D. Winter and D. Dodou. "Five-Point Likert Items: T Test versus Mann-Whitney-Wilcoxon". In: *Practical Assessment, Research & Evaluation* 15 (2010), p. 11. DOI: 10.7275/bj1p-ts64.

[194] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema. "Constraint Handling in Multiobjective Evolutionary Optimization". In: *IEEE Transactions on Evolutionary Computation* 13.3 (June 2009), pp. 514–525. DOI: 10.1109/TEVC.2008.2009032.

[195] C. S. Y. Wong, A. Al-Dujaili, and S. Sundaram. "Hypervolume-Based DIRECT for Multi-Objective Optimisation". In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. GECCO '16 Companion. New York, NY, USA: ACM, 2016, pp. 1201–1208. DOI: 10.1145/2908961.2931702.

[196] L. Yan et al. “An Interpretable Mortality Prediction Model for COVID-19 Patients”. In: *Nature Machine Intelligence* 2.5 (5 May 2020), pp. 283–288. DOI: 10.1038/s42256-020-0180-7.

[197] A. Yassine et al. “Information Hiding in Product Development: The Design Churn Effect”. In: *Research in Engineering Design* 14.3 (Nov. 1, 2003), pp. 145–161. DOI: 10.1007/s00163-003-0036-2.

[198] G. G. Yen. “An Adaptive Penalty Function for Handling Constraint in Multi-Objective Evolutionary Optimization”. In: *Constraint-Handling in Evolutionary Optimization*. Ed. by E. Mezura-Montes. Studies in Computational Intelligence 198. Springer Berlin Heidelberg, 2009, pp. 121–143. DOI: 10.1007/978-3-642-00619-7_6.

[199] B. Y. Yu et al. “Human Behavior and Domain Knowledge in Parameter Design of Complex Systems”. In: *Design Studies* 45 (July 2016), pp. 242–267. DOI: 10.1016/j.destud.2016.04.005.

[200] S. Zapotecas-Martínez et al. “A Review of Features and Limitations of Existing Scalable Multiobjective Test Suites”. In: *IEEE Transactions on Evolutionary Computation* 23.1 (Feb. 2019), pp. 130–142. DOI: 10.1109/TEVC.2018.2836912.

[201] Q. Zhang and P. N. Suganthan. *Final Report on CEC ’09 MOEA Competition*. Tech. Rep. UK: The School of CS and EE, University of Essex, 2009.

[202] Q. Zhang et al. “Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition”. In: *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report* 264 (2008).

[203] B. J. Zimmerman. “Self-Efficacy: An Essential Motive to Learn”. In: *Contemporary Educational Psychology* 25.1 (Jan. 2000), pp. 82–91. DOI: 10.1006/ceps.1999.1016.

[204] E. Zitzler and L. Thiele. “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach”. In: *IEEE Transactions on Evolutionary Computation* 3.4 (Nov. 1999), pp. 257–271. DOI: 10.1109/4235.797969.

Acronyms

AI	*artificial intelligence*
ASME	American Society of Mechanical Engineering
BLDC	brushless DC
CAD	computer-aided design
CDIO	conceive-design-implement-operate
CDP	constrained dominance principle
CDS	computational design synthesis
CHS	constraint handling strategy
CMOP	constrained multiobjective optimization problem
DAC	Design Automation Conference
DoF	degrees of freedom
EA	evolutionary algorithm
FCL	*flexible collision library*
FEM	finite-element method
FsR	feasability ratio, see Definition 3.1
FVC	fitness violation correlation
GAN	generative adversarial network
HVAC	heating, ventilation, and air conditioning
IGD	inverted generational distance
IPMQ	Interprofessional Project Management Questionnaire
IQR	interquartile range
IZ	ideal zone
JE	Johnson Electric
MODAct	Multi-Objective Design of Actuators
MOEA	multiobjective evolutionary algorithm
MOO	multiobjective optimization
nRFB$_\times$	normalized ratio of feasible boundary crossing, see Eq. (3.7)
ODE	ordinary differential equation
OMOD	Online MODelling platform
PBL	problem-based learning
PWM	pulse-width modulation
RFB$_\times$	ratio of feasible boundary crossing, see Definition 3.2

SD	standard deviation
SR	stochastic ranking
UEA	unbounded external archive

Cyril PICARD

EDUCATION

June 2021

PhD in energy (EDEY), EPFL, Switzerland
Automated Design: A Journey Across Modelling, Optimization, and Education

Supervisor: Prof. Jürg Schiffmann

Sept 2013 - Sept 2015

MSc in mechanical engineering with a minor in «Energy» (ing. méc. dipl. EPF), EPFL, Switzerland
Focus on control systems and mechatronics.

Master thesis supervisior of Dr. Christophe Salzmann, *Laboratoire d'Automatique*: *Non-destructive testing imaging system with real-time probe position acquisition*

Sept 2010 - July 2013

BSc in mechanical engineering, EPFL, Switzerland
Bachelor project: Building an **automated system to outperform a human player** using powerful motors and a high-speed camera (youtu.be/QQS0415YYW4).

WORK EXPERIENCE

June 2016 - June 2021
data-driven methods, machine intelligence, ray-tracing, evolutionary algorithms, mechanical design, teaching, teaching research

Doctoral assistant – Laboratory for Applied Mechanical Design, EPFL
Researching **data-driven modelling** methods and evolutionary **multi-objective optimization algorithms** for constrained problems with the aim to develop automted design tools and methodologies to design complex mechanical system. Conduct research with students to evaluate their learning of professional and computational thinking skills.

Exercise and project development and supervision for Bachelor and Master students from the mechanical engineering department.

Dec 2015 - May 2016
modeling of buildings for energy control, MPC, user feedback tools, Matlab, team work

Research assistant (civil service) – iHomeLab (HSLU), Luzern, Switzerland
Active on various research projets in the field of **energy efficiency and building intelligence**. Design of control strategies that reduce the energy consumption and promote load shifting (network stabilization) in service buildings and individual homes.

Feb 2015 - Oct 2015
Integrated mechanical design, embedded hardware and software design, Python, C, C++, VTK team work, git workflow

R&D Engineer – Master thesis – Sensima Inspection, Gland, Switzerland
Design of an innovative product to simplify the detection of defects in metal parts by delivering real-time 3D views. **Integration of a positioning system** with existing company hardware. Extension of the visualization software to **collect and process heterogenous data**.

Preparation of demonstration workflows and participation at the 2015 ASNT Annual Conference in Salt Lake City, UT, including handling discussion with potentiel customers.

Oct 2009 - March 2010
energy savings, child education, interaction with political institutions, project management

Project manager (civil service) - «terragir, énergie solidaire», Meyrin, Switzerland
Manager of the *Robin des Watts* project of the association «terragir, énergie solidaire» (terragir.ch): **awareness lessons for children on energy savings and sustainable development**. Design of specific training tools for teachers. Close collaboration with public administration in Geneva.

May 2007 - June 2007

Intern – Swiss Institute of Bioinformatics (SIB), Lausanne, Switzerland
Developement of a problem-specific database for the storage of theoretical and experimental results combined to a web interface coupled to bioinformatic tools from SIB to analyze and visualize the data. This project was under the supervision of Dr. Laurent Falquet in collaboration with the university of Basel.

AWARDS

- 2018 ASME Turbo Expo Best Paper Award (GT2018-76349)
- 2017 SIEGVO Summer School Best Assignment Award
- 2015 Bombardier Award for best grades during MSc
- 2013 venture kick: Stage 1 kick-off funding for potential entrepreneurs
- 2013 EPFL Excellence Fellowship for outstanding academic records during BSc
- 2009 Prix Marc Birkigt for best grades in mathematics and physics in high school

PUBLICATIONS

Picard, C. & Schiffmann, J. **Realistic Constrained Multiobjective Optimization Benchmark Problems From Design**. IEEE Transactions on Evolutionary Computation 25, no. 2 (April 2021): 234–46.

Picard, C. & Schiffmann, J. **Automated Design Tool for Automotive Control Actuators**. in IDETC-CIE2020 (American Society of Mechanical Engineers Digital Collection, 2020).

Mounier, V., Picard, C. & Schiffmann, J. **Data-Driven Pre-Design Tool for Small Scale Centrifugal Compressors in Refrigeration**. J. Eng. Gas Turbines Power (2018).

Picard, C. & Schiffmann, J. **Impacts of Constraints and Constraint Handling Strategies for Multi-objective Mechanical Design Problems**. in Proceedings of the Genetic and Evolutionary Computation Conference 1341–1347 (ACM, 2018).

Birrer, E., Picard, C., Huber, P., Bolliger, D. & Klapproth, **A. Demand response optimized heat pump control for service sector buildings**. Comput Sci Res Dev 32, 25–34 (2017).

Mathimaran, N. et al. **Unexpected Vagaries of Microsatellite Loci in Glomus Intraradices: Length Polymorphisms Are Rarely Caused by Variation in Repeat Number Only**. New Phytologist 180, 568–570 (2008).

Mathimaran, N. et al. **Microsatellites for disentangling underground networks: Strain-specific identification of Glomus intraradices, an arbuscular mycorrhizal fungus**. Fungal Genetics and Biology 45, 812–817 (2008).

Picard, C. & Picard, D. **A web application to manage a database for liquid nitrogen tanks**. Immunol. Cell Biol. 82, 257–259 (2004).

WHEN OFF-DUTY

- Volunteer at *Verein VoCHabular*, Zürich, Switzerland
- Convinced bicycle rider for transportation or vacations
- Amateur violin and viola player

Je maîtrise également le français. Wir können auch zusammen Deutsch sprechen. Schwiizertüütsch isch au e möglichkäit.