Multi-objective Multi-disciplinary Design Optimisation: from Benchmark Development to Real-world Applications in Public Health

Victoria Johnson

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

The University of Sheffield

Faculty of Engineering

Department of Automatic Control and Systems Engineering

March 2024

Victoria Johnson

Multi-objective Multi-disciplinary Design Optimisation: from Benchmark Development to Real-world Applications in Public Health

PhD thesis, **March 2024**

Supervisors: Professor Robin C. Purshouse
Professor Visakan Kadirkamanathan

The University of Sheffield

Faculty of Engineering

Department of Automatic Control and Systems Engineering

Amy Johnson Building

Portobello Street

Sheffield, S1 3JD

Abstract

Large and complex systems with multiple objectives and many constraints, components, and design variables requiring optimization are more prevalent now than ever before. Optimization of such complex systems is often approached as a single, large problem. However, the natural partitions of these systems, often associated with decision-making units, should be considered in research, industry, and organisations alike. Partitioning these systems into disciplinary sub-systems is a potential approach for the handling of the overarching optimization problem. One way that these problems are formulated is using a method called multidisciplinary design optimization (MDO). While a number of recent efforts have focused on MDO solutions, it is difficult to compare the effectiveness of these methods in the absence of a benchmark problem. In this thesis, first a scalable standardised MDO test problem is developed, based on a popular multi-objective optimization problem - the ZDT test suite. A multi-objective MDO framework is used to solve this problem, and then the results are compared with the original non-MDO test set. The problem is then expanded to a full test set, and the system topologies are varied. This benchmarking test set is then evaluated using a different MDO architecture. Following this, the multi-objective MDO framework is used to solve a case study from the public health domain. It is found in general that increasing the number of design variables and disciplines leads to slower convergence. In the scalable benchmark problems, when compared to the non-MDO problems, the solutions have a smaller hypervolume, meaning they are further away from the Pareto front. The main limitations in this research stem from the use of the ZDT test suite for the development of the multi-objective MDO problems, due to features such as separability and lack of scalability in objectives.

Declaration

I, Victoria Johnson, declare that the work presented in this thesis is my own. All material in this thesis which is not of my own work, has been properly accredited and referenced.

Signed: ____________________________

Victoria Johnson

Life before Death.

Strength before Weakness.

Journey before Destination.

– Brandon Sanderson, *The Way of Kings*

Acknowledgements

I would like to extend my deepest thanks and appreciation to the following people, without whom this PhD would not have been possible.

Firstly to my supervisor, Professor Robin Purshouse, whose kindness, patience and understanding saw me through the successful completion of this PhD. Thank you for the opportunity to work on this project, and the guidance that helped me make the most of it. Thank you to my second supervisor, Professor Visakan Kadirkamanathan, your suggestions and feedback were deeply valuable in this journey.

I extend my thanks also to Doctor João Duro. Your expertise has been invaluable in this endeavour.

Many thanks to the researchers at Greater Manchester Combined Authority who helped me formulate the problem in Chapter 8, and the researchers in the Systems Science in Public Health and Health Economics Research (SIPHER) Consortium who supported my work and gave me advice.

All my thanks to my family. To my fiancé Jamie, who believed in me and tolerated my ramblings. To my soon-to-be parents-in-law, David and Helen. The effect of your support cannot be overstated; it is not hyperbole to say that this PhD would not have happened without you. Thanks to my beloved family, my mother Sharon and my father Peter, for their unconditional love and understanding, and to Piper, who was always there. And finally, to my grandparents Anne and Jack who started this journey with me but sadly weren't able to see the end.

Lastly, thanks to the UK Research and Innovation (UKRI) Engineering and Physical Research Council for the financial support.

Contents

List of Figures

List of Tables

Acronyms

ATC Analytical Target Cascading.

CO Collaborative Optimisation.

CSSO Concurrent Subspace Optimisation.

CTP Constrained Test Problem.

DHI Disposable Household Income.

DMOEOA Disruption-Based Multiobjective Equilibrium Optimization.

DOE Design Of Experiments.

DTLZ Deb-Thiele-Laumanns-Zitzler.

EIQ Income Inequality.

EPR Employment Rate.

FPR Fuel Poverty.

GM Greater Manchester.

GMCA Greater Manchester Combined Authority.

HLE Healthy Life Expectancy.

IDF Individual Disciplinary Feasible.

IMD Index of Multiple Deprivation.

JPC Job Security.

LA Local Authority.

LAA Local Authority District.

LHS Latin Hypercube Sampling.

LSOA Lower layer Super Output Area.

MCDM Multi-Criteria Decision Making.

MCS Mental Health.

MDA Multidisciplinary Analysis.

MDF Multi-Disciplinary Feasible.

MDO Multidisciplinary Design Optimisation.

MDO-TTP Multidisciplinary Design Optimisation Travelling Thief Problem.

MO-MDO Multi-Objective Multidisciplinary Design Optimisation.

MOO Multi-Objective Optimisation.

MOP Multi-Objective Problem.

MSOA Middle layer Super Output Area.

NASA National Aeronautics and Space Administration.

NHS National Health Service.

NSGA-II Nondominated Sorting Genetic Algorithm 2.

NVQ National Vocational Qualification.

OA Output Area.

PLW Labour Remuneration.

PMR Premature Mortality.

PVT Poverty.

RSM Response Surface Model.

SAQ Skills and Qualifications.

SCC Sheffield City Council.

SII Slope Index of Inequality.

SIPHER Systems science In Public Health and Health Economics Research.

SLSQP Sequential Least Squares Quadratic Programming.

WFG Walking Fish Group.

XDSM eXtended Design Structure Matrix.

ZDT Zitzler-Deb-Thiele.

Chapter 1

Introduction

1.1 Background

Dynamic decision-making and systems science are closely linked disciplines that can be applied to a wide range of complex problems [18]. It has been noted that it is difficult to get a group of systems specialists to agree on the definition of a system [19]. For clarity, the definition of a real system, proposed Dori and Sillitto will be used here. A real system is '[t]wo or more elements interacting in physical space-time to create emergent properties, capabilities, functions or effects that the elements in isolation cannot achieve' [20].

Decision-making is often central to the use or analysis of these systems, particularly when there are important choices to be made regarding choices of structure, components, or trade-offs.

Decision-making in large systems may be particularly complex, involving many different preferences or requirements from actors or stakeholders in the system who have their own individual aims or goals, which may change as the system evolves through time. Social systems in particular may contain elements that, when perturbed, can create their own far-reaching disturbances that affect seemingly unrelated elements through a series of effects and processes [21]. It is effects like this that make systems science and dynamic decision-making difficult, yet interesting, topics to address.

As the needs of different organisations and engineered products change and become more tightly coupled, the optimization problems produced by such situations become much more difficult to solve using standard optimization approaches. The same goes for the complex problems found in social systems and devolved government, such as those concerning public health policy. These may include, but are not limited to, adjusting the price of alcohol [22] to healthy eating interventions [23]. Such policy problems are complex and multi-faceted, requiring an approach involving systems thinking and dynamic decisions making. Most importantly, they may benefit from disciplinary partitioning or decomposition.

Multidisciplinary design optimization (MDO) is one way to approach these types of problems.

MDO is a field of optimization that utilises numerical approaches to solve problems that contains more than one discipline, subsystem or component. Partitioning a system into multiple disciplinary subproblems may be logical or convenient because large, complex engineering problems often involve multiple teams with different specialisms — as such, MDO mimics the structure of industrial projects, lending itself to such problems. MDO is typically useful in (and was developed in response to) these settings because it silos the problem into multiple sub-problems or disciplines [24]. The earliest roots of MDO were in structural optimization [25], and as is noted in [26], the concepts used here are applicable to other fields of engineering, because in most engineering problems there are subsystems or disciplines that attach to a central structure.

Decomposition in MDO can arise for a number of interrelated reasons: from engineering practitioners taking a ‘divide-and-conquer’ approach to solving complex design problems, to the way in which engineering disciplines have emerged over time as discrete entities, to the functional organisation of teams in large engineering companies and institutions. Whilst MDO is useful in the design of complex engineered products, such as those that exist in aerospace [27, 28, 29, 30, 31], automotive [32, 33], marine [34, 35, 36, 37, 38] or robotic design [39, 40], its application is not limited to engineering, but is equally applicable to other complex systems contexts such as environmental and public policy [41, 42].

In MDO, disciplines interact with each other using *linking* (or *coupling*) variables, which allows other disciplines to solve their individual problems using information from the other components in the system. These interactions are important because the performance of a system is not necessarily defined just by its components, but also by the interactions between those components. However, when the subsystems have circular dependencies, it is not trivial to determine the values of the linking variables, and it might be necessary to use numerical approximation techniques, such as a multidisciplinary analysis (MDA) solver.

Multi-objective multidisciplinary design optimization (MO-MDO) extends the MDO problem by increasing the number of objectives in the problem, commonly using techniques found in multi-objective optimization literature. This includes a priori, a posteriori, and interactive techniques, performance indicators, optimization engines, benchmarks, and so on. MO-MDO has comparatively little dedicated research (especially when compared to the research in evolutionary multi-criterion optimization (EMO), which is a very large field), but there is a growing body of research on MDO problems with multiple objectives [43, 44, 41, 45, 39]. Once again, most of this research has applications in industrial settings.

MO-MDO is an important and natural extension of the single-objective MDO research because many of the 'real world' problems found in industry, public health and operational research, involve multiple objectives and trade-offs. Therefore, it is important to include the ability to include such features in MDO strategies, benchmarks, and research.

However, a thorough search of the related literature has revealed that MO-MDO has only once been applied to problems found in public policymaking, such as those faced by devolved authorities [41], and has not been applied to a problem in collaboration with these organisations. MO-MDO is potentially a good option for tackling the type of problems found in this area, because devolved and national authorities are typically composed of smaller departments with their own objectives, constraints and variables. An obvious example of this can be seen in the UK Government, which, in March 2024, has 24 ministerial departments, 20 non-ministerial departments, and over 400 agencies and public bodies [46].

In this project, MO-MDO techniques are researched and applied to a real-world social problem.

1.2 Motivation

Dynamic systems such as devolved authorities with large collective numbers of people are notoriously difficult to model, simulate, and optimize. For example, changes in housing can affect public health, which can also affect productivity, which can affect the distribution of jobs, which can then go on to further affect housing again. Small actions in one area can create large reactions in other, seemingly unrelated areas. To strike the balance between actions, reactions, and priorities in the above council and government areas, MO-MDO can be used to inform and improve policy for these councils, in settings where decisions are distributed across different functional areas (for example, in housing or transport).

The key motivation for this project is to bring MO-MDO approaches and strategies into the area of public policymaking, for use by devolved governments when aiming to optimize the amount of funding spent for the best possible outcomes.

1.3 Aims and objectives

The aim of this project is to develop a framework for dynamic decision-making systems that can be benchmarked on a standardised problem suite. It is then to be applied to a case study in public health.

The objectives were developed to outline a series of steps to achieve this aim:

1. Develop an MO-MDO formulation based on the ZDT1 multi-objective optimization problem as a framework for handling scalable numbers of disciplines.

2. Create a full suite of the MDO-ZDT benchmark test set on which to evaluate MO-MDO optimizers and architectures.

3. Introduce a bi-level MO-MDO architecture and include constraints in the distributed architecture for the multi-disciplinary optimization problem.

4. Develop a multi-criteria decision-making (MCDM) problem formulation suitable for informing local government policies. This should include decision variables, criteria, and stakeholder preferences to relate the problem to how local government represents problems.

1.4 Thesis overview

Chapter 2 contains an overview of the key concepts in both multi-objective and multidisciplinary design optimization. These key concepts are important for understanding the work in the later chapters. Additionally, this chapter covers some of the important or commonly used benchmarks in MDO, MOPs, and the current status of benchmarking problems in MO-MDO.

Chapter 3 is a review of the state-of-the-art in methods in MO-MDO. Primarily, these are grouped into the a priori and a posteriori methods, and then further separated by architecture. Chapter 3 contains a typology for identifying whether the methods in the literature are multi-objective at the system, subsystem, both, or some combination of these.

Chapter 4 first gives a short outline of the research choices made in the thesis, including the chosen benchmarks from which the MO-MDO benchmark is developed, the optimization engine, and the MDO architectures. Following this, Chapter 4 proposes a bi-objective MDO problem using the MDF architecture based on a combination of the ZDT1 problem [47] and the scalable problem proposed by Tedford and Martins [48]. Scalability of the problem is tested for two, seven and 14 disciplines. This problem is implemented in the Python programming language.

Chapter 5 is an extension of the work in Chapter 4. Three main improvements are proposed. Firstly, the multidisciplinary test problem applied to only

ZDT1 in Chapter 4 is extended to all continuous problems in the ZDT test suite. Secondly, the linking relationships between the disciplines are expanded and made more complex. Three potential relationships are defined. Thirdly, a term is added to the problem that penalises 'incorrect' linking variable values.

Chapter 6 modifies the MDO-ZDT problems proposed in Chapters 4 and 5 and applies them to a Collaborative Optimization architecture. An additional modification is that constraints are added at the disciplinary levels. Optimization convergence and similarity to the original ZDT Pareto front are studied.

Chapter 7 applies the MO-MDO frameworks developed and studied in Chapters 4 to 6 to a real-world public health inequality problem in a combined local authority in England.

Chapter 8 provides a summary and conclusion of the work in the thesis, as well as outlining future research directions.

1.5 Original contributions

The following original contributions have been made during the PhD.

1. A review of the methods and strategies in MO-MDO has been written. This includes a priori and a posteriori approaches applied to a range of different MDO architectures. This is planned for submission to a journal for publication.

2. A scalable MO-MDO problem was developed based on the ZDT1 multi-objective problem, and the effect of varying the number of design variables and disciplines was studied. This contribution was published in the following conference paper.

   V. Johnson, J. A. Duro, V. Kadirkamanathan and R. C. Purshouse, "Toward scalable benchmark problems for multi-objective multidisciplinary optimization," 2022 IEEE Symposium Series on Computational Intelligence (SSCI), Singapore, Singapore, 2022, pp. 133-140, doi: 10.1109/SSCI51031.2022.10022207.

3. A set of scalable MO-MDO problems was developed by extending the problem

in point (2). The topologies of these problems was varied, and the outcomes were analysed. This contribution was published in the following conference paper.

V. Johnson, J. A. Duro, V. Kadirkamanathan, and R. C. Purshouse. 2023. A scalable test suite for bi-objective multidisciplinary optimization. In Proceedings of the 2023 International Conference Series on Evolutionary Multi-Criterion Optimization (EMO). 319–332.

4. The MDO ZDT test problems developed in points (2) and (3) were then applied to a distributed MDO architecture called collaborative optimization. The number of disciplines and constraints was varied for each of the continuous problems and the results were analysed. This contribution was published in the following conference paper.

   V. Johnson, J. A. Duro, V. Kadirkamanathan, and R. C. Purshouse. 2023. A distributed multi-disciplinary design optimization benchmark test suite with constraints and multiple conflicting objectives. In Proceedings of the Companion Conference on Genetic and Evolutionary Computation (GECCO '23 Companion). Association for Computing Machinery, New York, NY, USA, 1611–1619. https://doi.org/10.1145/3583133.3596414.

5. The final contribution is the development of an optimization problem focused on reducing health inequalities and maximising average mental health in a combined authority in the UK. The concepts developed and studied in the previous chapters are applied here to formulate and solve the problem. It is intended that this study will be developed into a more advanced and realistic problem following the PhD, and that the researchers or policy makers in combined or devolved authorities can use this research to inform their own public health policies.

Chapter 2

Background, key concepts and benchmarks

In this chapter, the key concepts for both MDO and multi-objective optimization will be presented and discussed in Section 2.1, including the fundamental concepts for MDO and MOP research such as different variable types, architectures, MOP formulation and the concept of dominance. Following this, some of the most important benchmark problems in MDO, MOP and MO-MDO research are presented in Section 2.2, including both analytical/academic benchmarks, and those that arose from real-world applications or engineering design problems. Section 2.3 discusses some of the experimental choices made through this thesis, considering some of the points made in Sections 2.1 and 2.2. Section 2.4 gives an overview of the chapter and introduces the following chapter.

Figure 2.1 gives an overview of the contents of this chapter.

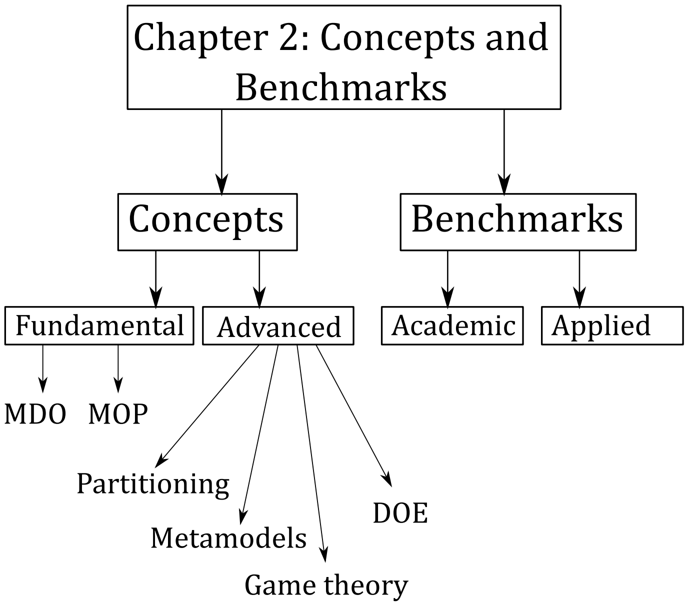


Figure 2.1: The structure of Chapter 2, covering concepts in MO-MDO and benchmarks.

2.1 Key concepts

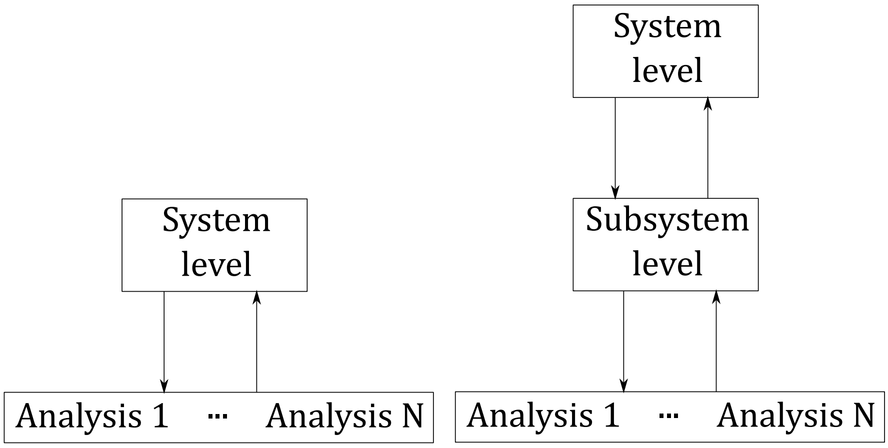
2.1.1 Fundamental concepts

MDO and MDO concepts

Multidisciplinary design optimization (MDO) has been applied in many different industries, including design of aircraft and aircraft parts [49, 28, 50, 29, 51, 52, 53, 54], heavy machinery [55, 56], production planning [57], and various smaller test problems [58, 59] (also see [60] for a set of test MDO problems developed by NASA - problems such as the Golinski speed reducer are very common test cases in MDO research [61]).

There are some key concepts in MDO that must be specified before going into further detail about MO-MDO.

Systems and subsystems The *system* or system-level problem in MDO is the top-level optimization or coordination problem from which all other sub-optimizations and analyses are initiated. In the system level, there will typically be an optimization problem complete with objective functions, constraints, and bounds. All information



(a) Monolithic architectures (b) One example of a distributed architecture

Figure 2.2: Simplified schematic for monolithic and distributed architectures.

from sub-levels are drawn on here to reach a solution or set of solutions. The *subsystem* problem is below the system-level problem in the MDO hierarchy. Subsystem optimization problems are only present in distributed architectures. A recomposed distributed problem will be the same as a corresponding monolithic problem. *Analyses* are the equations or models that are invoked either by the system or subsystem optimizers and represent the disciplines. These are solved by multidisciplinary analysis (MDA) – in some architectures, this simply means returning the evaluation of the analyses to the corresponding optimizer, while in other architectures, the MDA solver is responsible for ensuring all solutions are consistent with each other. See Figure 2.2 for an explanatory diagram of monolithic and distributed architectures.

Variables In MDO, there are three different types of *variables*: global variables (referred to in this thesis as z), local variables (x_i) and linking variables (y). Global variables are sometimes known as shared variables [24] – these are accessible by all modules within the MDO formulation, including all subsystems and the system-level problem. Local variables are only accessible by their corresponding subsystem and the system-level functions such as objectives and constraints. Linking variables are generated by subsystem analyses and are sent to and used by other subsystems, either by the MDA solver, or by the system level. Linking variables and global variables are accessible by more than one discipline, while local variables are not.

In MDO, disciplines interact with each other using linking variables (sometimes known as coupling variables), which allows other disciplines to solve their individual problems using information from the other components in the system. In some architectures, these disciplines may have circular dependencies, making such systems of problems difficult to solve.

Architectures In MDO, the *architecture* is the combination of processes, steps, or connections that make up the optimization process. As stated in [24], there are *monolithic* and *distributed* architectures; in monolithic architectures, there is one optimization problem with an analysis module (also referred to as a solver) that evaluates the outputs of the model. In some monolithic architectures, the solver also handles the linkages between the subsystem analyses, such as multidisciplinary feasible, where the feasibility and consistency of the evaluation outputs must be resolved before the optimization problem. In other architectures, the solver acts only as an evaluator of models. For example, this happens in individual disciplinary feasible architectures, where the consistency between disciplines is resolved by the optimizer. In distributed architectures, the optimization problem itself is decomposed into more than one subsystem, which also have their own optimization problems (and a solver for their corresponding analyses).

There are many MDO architecture in the relevant literature, but four will be included here because of their relevance to the literature review and the research contained within the thesis. The full formulations are shown in Appendix A.

XDSM diagrams XDSMs are an extension of the design structure matrix (shortened to DSM) [62], which use matrices to represent constituent components of a large system. This technique has been popularised by [24] to visualise the interconnections between the components of a complex system. It is useful in particular to visualise both data dependencies and process flow. The disciplines, analyses and optimizations (referred to collectively here as *components*) are represented on the diagonal by oblongs or rectangles, the input data flows on the vertical direction, and the output data flows on the horizontal direction. Data are labelled inside grey parallelograms, and the way the data flows is shown as thick grey lines. The black lines represent connections between data and processes. In some cases, the components, inputs, outputs, or data boxes appear to be 'stacked'. This means that there are more than one of these components, inputs, outputs, or data boxes working in parallel. For example, if there is a stacked rectangle box labelled 'system analysis i', this means that there are several system analyses being executed simultaneously.

Throughout this thesis, XDSMs will be used to illustrate the architecture of methods discussed, where appropriate. The paper giving further details on how these diagrams were developed and how to read them are found in [63]. The XDSMs were generated using the pyXDSM package [63].

Multi-objective optimization concepts

In multi-objective optimization (MOO), there is more than one objective function. The body of research on MOO is large and covers many key areas such as methods and strategies, initialisation and design of experiments, optimization engines, trade-offs, and how to involve the decision maker in the process. For brevity, only the most relevant concepts will be covered here.

Formulation The generic, constraint-free multi-objective optimization problem can be expressed as

$$\begin{aligned}
\text{minimise} \quad & F(x) = [f_1(x), f_2(x), ..., f_n(x)] \\
\text{subject to} \quad & x \in \mathcal{D} \\
& g_i(x) \leq 0, \qquad i = 1, ..., n_g \\
& h_i(x) = 0, \qquad i = 1, ..., n_h \\
& x_L < x < x_U \\
& \mathcal{D} \rightarrow \mathbb{R}^k
\end{aligned} \tag{2.1}$$

where x is the design vector, $\mathcal{D}$ is the decision space, $n > 1$ is the number of objective functions to be minimised, $F(x)$ is the vector of objective functions $f(x)$ to be minimised, $g_i(x)$ is inequality constraint i, $h_i(x)$ is equality constraint i, and

n_g and n_h are the total number of inequality and equality constraints. x_L, x_U are the upper and lower bounds on the design variables. Objective functions requiring maximisation can be represented as minimising the negative of the objective function, that is, $\max F(x) = -\min F(x)$.

Trade-off and domination In a problem with more than one conflicting objective, there will always exist a trade-off. A continuous design space will harbour infinite optimal solutions, and it falls to the decision-maker to choose which design to implement. A Pareto optimal solution is a solution to an optimization problem where no improvement can be made to one objective without damaging the performance of another, meaning that the aforementioned trade-off has to occur. Mathematically, any point $\boldsymbol{x_1^*} \in \mathcal{D}$ is Pareto optimal if and only if $\{\nexists \boldsymbol{x_2^*} \in \mathcal{D} | f_i(\boldsymbol{x_1^*}) \leq f_i(\boldsymbol{x_2^*}) \forall k \vee f_i(\boldsymbol{x_1^*}) < f_i(\boldsymbol{x_2^*}) \exists i\}$ [64]. Consequently, the decision-maker will choose which design is the most appropriate solution, while incorporating goal preferences.

A design is dominated if there is some other feasible design that outperforms the initial design in all objectives. Similarly, a non-dominated design is one that is not outperformed in any objective by some other design. Mathematically, for two designs $\boldsymbol{x_1^*}$ and $\boldsymbol{x_2^*}$, $\boldsymbol{x_1^*}$ is non-dominated if $\{\forall \boldsymbol{x} \in \mathcal{D} | f_i(\boldsymbol{x_1^*}) < f_i(\boldsymbol{x_2^*}) \forall i\}$. These non-dominated designs are known as the approximate Pareto front [65]. An example of an approximated Pareto front is shown in Fig 2.3.

2.1.2 Advanced concepts

Problem partitioning

Partitioning is when a large optimization problem is decomposed into multiple subproblems, often organised by discipline or component. In some applied industrial problems, partitioning may occur according to design team, with each design team taking on a segment of the decomposed problem. For example, in the spacecraft engineering design problem found in [66], the aerospace system is decomposed into six disciplines: orbit, thermal, power, attitude, structure and propulsion, according

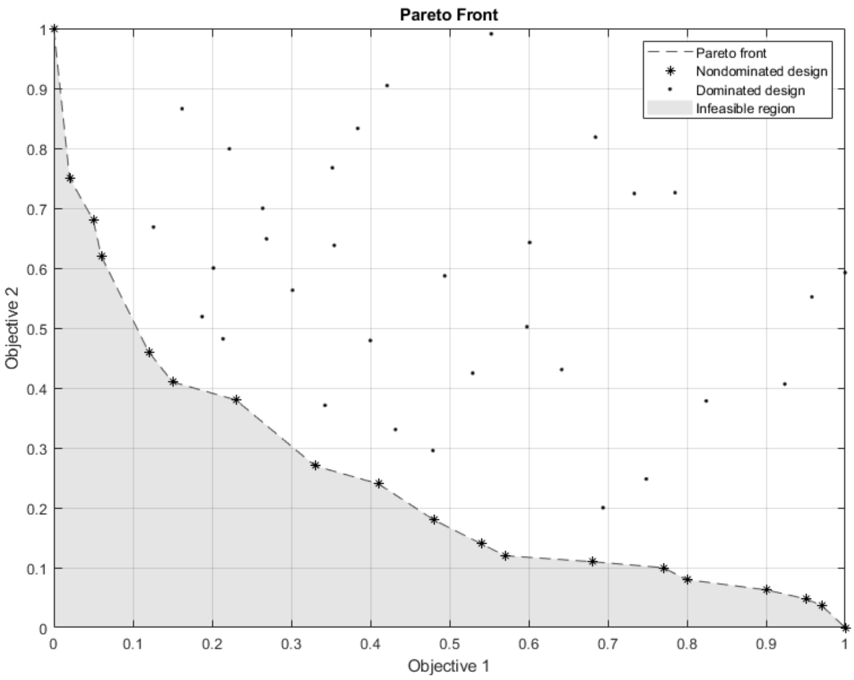


Figure 2.3: Example Pareto front in the decision space with designs evaluated for objectives 1 and 2.

to how the different disciplines interact, and the specialities of the engineers and analysts working on the craft.

Depending on the problem at hand, MDO problems may either come pre-partitioned (in the case of some academic benchmark problems) or need partitioning manually. The way the problem is partitioned into subproblems, shared and local variables, can have a substantial impact on the overall outcome of the optimization process.

Four partitioning strategies are commonly cited, presented by Wagner and Papalambros [67] and mentioned in [68]: model-based, sequential-based, aspect-based and object-based.

In model-based partitioning, an algorithm decides the partitions of the system. Model-based partitioning often makes use of graph and matrix representation of systems [69]. This type of partitioning is different to sequential-, aspect- and object-based partitioning because it may rely only on the mathematics of the problem, rather than being influenced by outside aspects such as workflow, components, or the structure of the business or institution engineering the complex system in question. Tosserams et al. have developed a software called Ψ that facilitates the partitioning of such problems [70]. Additionally, Rogers and Bloebaum developed a software for the partitioning of large systems called Design Manager's Aid for Intelligent Decomposition (DeMAID) [71], which also identifies the couplings that have low or no impact on the outcome of the optimization process, leading to their removal for computational efficiency.

Graph-based partitioning has also been proposed as a variant of model-based partitioning [72, 73, 74], which uses Boolean matrices to represent the relationships between design variables and objective and constraint functions. These are referred to as *functional dependence tables* and can also be expressed as hypergraphs, in which the nodes (objectives, constraints) are linked by edges (design variables). Candidate global, linking and local variables can be identified using this method, and the hypergraphs can be optimized for various metrics, such as edge weight. This is done using a method called optimal model-based decomposition.

Sequential-based partitioning is organised according to the workflow of the problem or engineered product. In this approach, it is assumed that information flows in strictly one direction. Note that sequential partitioning is not mentioned in [70]. Sometimes sequential partitioning is used to inform a model-based partitioning algorithm, as in [75, 76, 77].

Object-based partitioning occurs according to the physical components of the problem [67, 70, 68]. Linking variables may be inputs or outputs to or from the system components.

Aspect-based partitioning is when the problem is divided according to specialities or expertise [67, 70, 68]. An example of this is in the robotic fish system proposed in [39], where the disciplines are hydrodynamics, propulsion, equilibrium, and energy management.

Metamodels

Metamodels, also known as surrogate models, are simplified models of models. In other words, when a model is too complex, computationally expensive or time-consuming to run within a given context, a metamodel is employed to reduce the burden posed by the original model. In some MDO architectures such as MDF or IDF, models may be run in the MDA module many hundreds or thousands of times before an optimal (or even feasible) solution is obtained, meaning that a little time saved on the execution of one model will save a lot of time over the optimization process.

Response surface methodology (RSM) is particularly popular in the MO-MDO literature (see for example [78, 30, 79, 80]). RSM uses designed experiments and statistics to model a surface that approximately maps the relationship between different explanatory variables and a model outcome.

Polynomial interpolation [30, 81], neural networks [14], Kriging interpolation [81, 54, 82, 83, 33] and gene expression programming [79] have also been used in the MO-MDO literature. In [66], multiple different types of models are reviewed for use in MDO, such as polynomial RSM, radial basis functions, Kriging interpolation, ensemble method, and machine learning-based methods. Also reviewed are multi-fidelity models, which make use of high- and low-fidelity data such that the accuracy of the model output can be high, while minimising use of computational resources, such as co-Kriging.

Game theory

Game theory is a mathematical framework that facilitates the modelling of interactions between entities in strategic situations (these are commonly referred to as players in a game).

It can be argued that game theoretic approaches lend themselves naturally to MDO problems because the multi-disciplinary structure of MDO problems can be considered analogous to the players in a game. The communication of system and subsystem solutions and linking variables can be seen as information-sharing (or lack thereof) between game players, and therefore, game theoretical approaches are a common approach to handling MDO problems.

In 1997, Lewis and Mistree noted the potential for treating subsystems as players in a game, proposed an approach to handling MDO problems that made use of common game theoretical strategies [78]. While the approach proposed in this paper does not adhere to any specific classical MDO architecture, it is important to note anyway because subsystems control their own local variables and interact with each other according to their game strategy. Acknowledging that not all design variables can be controlled by each subsystem (or player), three strategies can be used to handle the interactions between each subsystem: Pareto (cooperative), Nash (non-cooperative) and Stackelberg (leader-follower). Each subsystem is given one objective to minimise.

In Pareto cooperation, there is full communication between all subsystems, so there is no need to anticipate how the other subsystems will act, but in non-cooperation or approximate-cooperation, it is necessary for each subsystem to construct a model of how they expect the other subsystem to act or react; these models are called rational reaction sets (RRS), which is a formal expression of the assumptions each subsystem makes. Sometimes, if the problem is simple, exact RRSs can be obtained, but in most complex problems, they have to be approximated via surrogate or metamodels.

The test case used in [78] is an aerospace engineering problem with two disciplines: aerodynamics and weights. It is found that both full- and approximate-cooperative design produces the best overall solutions to the problem, and non-cooperative solutions perform worse. It is acknowledged that the accuracy of the RRS heavily influences the result in non-cooperative architectures. Additionally, increasing the number of variables used in a discipline's RRS drastically increases the complexity of each metamodel, therefore pruning of variables should be performed where possible, to reduce the dimensionality of the RSM and design of experiments (DOE).

Xiao et al made use of a game theoretical approach to MDO problems [79], noting that the cooperative game is analogous to the MDF architecture, while the non-cooperative game is analogous to most distributed IDF architectures such as CO. This is because subsystems do not communicate variables with each other in IDF-based architectures, and instead send and receive variables directly from the system-level problem.

Duro et al. [13] also use the Stackelberg leader-follower strategy in their paper on MO-MDO. More information can be found in Section 3.3.

In a similar approach, Ghotbi and Dhingra apply a multi-level, game theoretic approach to a flywheel design problem with two objectives and a Stackelberg strategy [84]. Different variable partitions are considered based on leader-follower configurations, and sensitivities from the follower problem are used to update the follower variables in the leader problem.

Design of experiments, variable pruning and sensitivity analyses

Some of the above meta-modelling methods such as RSM rely on a suitable design of experiments to obtain the most accurate possible metamodel whilst minimising the computational efforts to obtain it. There are many DOE plans, including Latin hypercube designs [85] (for example used in [66]) and their variants, which are useful for problems of high dimensionality. The quantity and variation of DOE stretches far beyond Latin hypercube sampling (LHS) schemes, but are out of scope for this literature review.

Variable pruning is also an important aspect of design of experiments because, in models with very large numbers of variables, some variables may have more or less of an impact on the model outcome than others, and it is important to minimise the complexity of the model within reasonable bounds. As discussed in Section 2.1.2, some softwares such as $\boldsymbol{\Psi}$ [70] and DeMAID [71] interrogate the model to find out which linkages (linking variables) have minimal or no impact on the outcome of the analysis. These can then be pruned.

Global sensitivity analysis can be used to identify variables that contribute more and less to the model output, as well as identify optimal regions, ensure model robustness, and validate the model [86]. Some sensitivity analyses such as sample-based approaches rely on good design of experiments to obtain high quality samples, and therefore accurate identification of high- and low-impact variables. Local sensitivity analyses also occur in many of the standard MDO architectures because they provide the optimizers and solvers with information about the behaviour of the MDAs in the local vicinity of the analysis outcomes, advising where the optimization can be carried out to obtain better designs. In the case of high-fidelity, high-dimensional models, this can be extremely computationally expensive and time consuming. In architectures such as MDF, partial differential equations are used to conduct the local sensitivity analyses, effectively acting as a one-at-a-time sensitivity analysis. In other architectures such as classic CSSO, global sensitivity analyses are part of the optimization process [87, 24].

2.2 Benchmarks

2.2.1 Introduction to MOP and MDO benchmarks

MOP benchmarks

Test multi-objective problems (MOPs) are widely studied and abundant among existing literature in multi-objective optimization. They are important in the field of optimization because, in real-world problems, there is rarely only one objective to optimize – there is usually a trade-off in complex engineering design, whether in performance, cost, component weight, and so on. MOP benchmarks allow researchers and practitioners to evaluate in detail the performance of optimization algorithms before deployment on other problems.

Different MOPs and MOP sets have different features. The Zitzler-Deb-Thiele (ZDT) [47] test set contains six different problems, one of which has discrete variables, and is scalable in the number of design variables but not objectives and contains no constraints. The Deb-Thiele-Laumanns-Zitzler (DTLZ) [88] benchmarks contain seven problems and are scalable in both the number of design variables and objectives but contain no constraints. The Walking Fish Group (WFG) [89, 90] problems are scalable in variables and objectives, and the problems can be reconstructed according to the needs of the user doing the benchmarking. However, again, these contain no constraints. The bi-objective Black Box optimization Benchmarking (BBOB-biobj) [91, 92] constructs test problems out of multiple single-objective problems, but the Pareto fronts for these are unknown. Other test problems, such as the Constrained Test Problems (CTP) [93] place a greater emphasis on constraint satisfaction. Again, CTP is customisable via user-set parameters, but these are not scalable in number of objectives.

MDO benchmarks

MDO benchmarks are generally not as well studied as MOP benchmarks and are usually constrained to single-objective problems or applications to real problems. The NASA test suite, summarised in [60, 94] is one of the oldest and most well-known set of problems, and includes the Golinski speed reducer [95, 96], propane combustion problem, and the heart dipole problems [60, 94]. However, the benchmarks found in the NASA test suite do not adequately reflect the state of MDO research today due to their lack of conflicting objectives, low complexity, and non-scalability of disciplines, design variables, and simple inter-disciplinary relationships.

Tedford and Martins produced a study in 2010 benchmarking the performances of different MDO architectures using a few of the existing MDO test problems, including the propane combustion and Golinski speed reducer problems mentioned above [48]. They also used the two-discipline Sellar problem from 1996 and the scalable problem; these allow the number of disciplines, design and linking variables, and linking strength to be varied.

Largely, benchmarking in MDO is an under-studied area, especially when compared with the success of benchmarking in MOPs. As a result, MO-MDO, a sub-field of MDO, also has very little in the way of useful test problems.

2.2.2 Academic benchmarks

In this section of the review, academic benchmarks are defined as problems that are easy to implement from scratch in languages or software that are accessible for free (no proprietary software is required to run the optimization problems), are used as standard benchmarks across multiple papers, or are based on existing benchmark problems.

Most of the multi-objective MDO benchmark problems are based on single-objective MDO problems with modifications to make them multi-objective. However, some [59, 58] have been recently developed that make modifications to multi-objective problems to produce a multi-disciplinary problem.

Multi-objective Sellar problem

The original MDO Sellar problem (sometimes referred to as the analytical problem [48]) was proposed in 1996 [80]. It contains two disciplines with two linking variables, two global variables and one local variable which is used in discipline one. The problem also contains two constraints. The single-objective Sellar problem is a popular choice for benchmarking in the MDO literature [97, 60, 98, 48, 99], possibly because of its simplicity.

The Sellar problem was modified to accept multiple objectives in [8] by splitting the original objective function in two, with one objective function maximising the linking variable from discipline two. As in the original Sellar problem, there are two disciplines. The problem, in isolation from the CO-based formulation, is expressed as

$$\begin{aligned} \text{minimise} \quad & f_1 = x_1^2 + z_2 + y_1, \quad f_2 = -y_2 \\ \text{s.t.} \quad & \frac{y_1}{8} - 1 \geq 0 \qquad \frac{y_2}{10} - 1 \geq 0 \\ & -10 \leq z_1 \leq 10, \quad 0 \leq x_1 \leq 10, \quad 0 \leq z_2 \leq 10 \end{aligned} \tag{2.2}$$

The subsystem 1 MDA is

$$y_1 = z_1^2 + x_1 + z_2 - 0.2y_2 \tag{2.3}$$

and the subsystem 2 MDA is

$$y_2 = z_1 + z_2 + \sqrt{y_1} \tag{2.4}$$

As with the single-objective Sellar problem, the use of this test problem extends only to basic architecture testing and demonstration due to the low complexity and size of the problem. The multi-objective Sellar problem presents no real challenge to the optimizer; however, the Pareto front of the problem is easily recovered.

Multi-objective Golinski speed reducer

The original speed reducer was proposed by Jan Golinski in [96]. The multi-objective version of this problem has been adapted directly from the NASA test suite [60, 94] in several different studies, making it a popular candidate for creating MO-MDO problems, with multiple uses in the literature [100, 11, 101, 102, 48, 103, 9, 104, 105]. The speed reducer is a gearbox that contains two gears, shafts and bearings; the seven design variables control various parameters, such as the width of the gear face (z_1), the teeth module (z_2), number of gear teeth on pinion (z_3), distance between bearings 1 and 2 ($x_{1,1}$, $x_{2,1}$) and the diameter of the shafts ($x_{2,1}$, $x_{2,2}$). All design variables are continuous, except z_3, $x_{2,1}$ and $x_{2,2}$, which are discrete. The variable notations have been changed slightly here from how they are presented in the paper – the standard MDO notation is used in this paper with z representing variables that are used across multiple subsystems and x_i representing variables that are local to subsystem i.

The aim of the single-objective problem is to minimise the weight or volume of the gearbox while satisfying constraints such as the stress of the gear teeth and the shafts, and the dimensionality constraints.

A bi-objective version of the problem was proposed in [100] where the aim was to minimise the volume of the gearbox (f_1) and the stress (f_2). This was not explicitly modelled as an MDO problem, but later iterations of the problem give an indication that the volume and stress could be separated into different subsystems [102].

A two-discipline distributed variation of the problem was proposed in [102], with two objective functions for each of the disciplines, which are to minimise the volume ($F_{i,1}$) and stress (f_i) on shafts one and two respectively. The volumes of the shafts are summed at the end of the optimization $F = F_{1,1} + F_{2,1}$, essentially making the problem tri-objective. This particular formulation does not contain an objective function at the system level, but does contain constraints and global variables.

Less-researched multi-objective NASA test problems

All of the problems in this section are from the NASA test suite but have not enjoyed the same amount of popularity as the Golinski speed reducer. A search of the literature only revealed one article that has created and used multi-objective versions of these [103], so they are placed under one subsection.

The heart dipole function is one of these problems [60, 94]. The test problem is originally stated to have eight objective functions, but the MDF and CO formulations in [94] uses a sum of four of them to produce one large objective function. Du and Simon reformulated this into a bi-objective problem with five constraints and six design variables [103]. However, as stated in their paper, this reformulation means the objective functions no longer have any physical meaning as they did before. Nevertheless, like the Golinski speed reducer, this problem is separable, which means it has good potential for use as a MO-MDO benchmark problem.

The power converter is another test problem. This problem consists of two subsystems – the electrical subsystem and the loss subsystem. The original NASA formulation has four constraint functions, six design variables and six linking variables, but only one objective function, which is to minimise the component weight [94]. Du and Simon reformulate this into a problem with two objective functions (the weights of the primary and secondary windings) [103]. Unlike some of the other converted multi-objective MDO problems with origins in the NASA test suite, this problem does not contain an objective made of a sum of smaller objective functions.

Finally, the propane combustion problem is the last in the list of less-researched multi-objective NASA test problems. This represents a mathematical model of the chemical equilibrium reached when propane combusts in air [48]. Like the heart dipole problem, the original formulation has many objectives (11 in this particular case), but the MDO formulation uses four to construct an objective function that is the sum of four of them [60, 94]. Du and Simon reformulate this problem into a tri-objective problem with eleven design variables, two disciplines, and four constraints [103].

Integrated location problem

The integrated location problem aims to optimize the placement of an airport and a hospital in a hypothetical town that serves the same set of people [41]. It is stated in the problem that the airport is repulsive to the hospital due to the noise, but the hospital is attractive to the airport because emergencies must be able to travel from the airport to the hospital in a short amount of time. A third facility which is to be used by both users of the airport and the hospital (the example in the paper is given as a childcare facility) must also be placed.

As a result, each of the subsystems have two objective functions: the maximisation of the distance from the hospital to the airport or the minimisation of the distance from the airport to the hospital in subsystems one and two respectively, and the minimisation of the distance from both the hospital and the airport to the childcare facility. The location of each of these places, the local variable x_i, acts as the linking variable to be sent to the other subsystem.

This problem is also solved in [13] using a CO-style architecture.

2.2.3 Applied problems

In this subsection of the review, applied problems are defined as those that do not conform to the definition set out in subsection 2.2.2. These are usually problems that have only been used in one paper or are difficult to implement in free-to-use software or languages.

Travelling thief problem

Along with the integrated location problem, also in [41] is a multi-objective MDO version of the well-known travelling thief problem [106] (MDO-TTP), which in itself is a combination of the knapsack problem [107] in the first subsystem, and the travelling salesman problem [108] in the second subsystem. In this problem, a thief is presented with a set of items to pack into his backpack, which has a finite volume for stolen items. Additionally, he has to visit every city within another set, finishing in the same city he started in. These subsystem problems interact via the relationship between the weight of the knapsack and the thief's speed – the heavier the knapsack, the slower the thief moves. The longer the thief takes to return to the original city, the lower the values of the items in the knapsack. Unlike most other problems discussed in this literature review, this is a combinatorial problem.

The local design variables are the items picked by the thief in subsystem one, and the route taken from city to city in subsystem two. The two linking variables represent the speed of travel when leaving each city and the value of each item.

This problem is slightly more complex than the Sellar problem because the multidisciplinary analyses are a little more complicated, making it a good choice for testing basic architectures. Structurally, however, the TTP is simple because the problem contains only two subsystems, two design variables, and no global variables. This problem has potential to be as widely used as the Sellar problem or the multi-objective NASA test suite – especially as the MDO-TTP is set up explicitly with the intention of use in multi-objective problems.

Propulsion problem

A multi-objective approach was used in 2015 for the design of a spacecraft propulsion system using the MDF and CO architectures, using SLSQP for the disciplinary optimizer and genetic algorithms for the system optimizer [43]. Surrogate models are used for the nine multidisciplinary analyses, which include combustion, the thrust chamber, nozzle geometry, and others. The number of design variables was reduced from 'a large number' to 13 for the sake of computational simplicity. The MDF problem contained two objectives and was optimized using the NSGA-II algorithm, with the aim of minimising wet mass while maximising the total impulse. In the CO formulation, two disciplines were used, each containing a substructure of five subdisciplines and one objective function each. Each objective function was a sum total of either the weighted impulse or wet mass, and a consistency objective J, which is in standard use in CO formulations. Once again, NSGA-II was used in the system-level problem, while SLSQP was used in the disciplinary super-problems.

Robotic fish design

The robotic fish design case study, presented in [39], proposes a multi-objective IDF approach with combinations of first principles equations and surrogate modelling in the multidisciplinary analyses. The problem contains four disciplines: hydrodynamics (surrogate MDA using a backpropagation neural network), weight and equilibrium (first principles equation MDA), propulsion (first principles equation MDA), and energy (first principles equation MDA) as well as 23 variables, which includes 14 design variables. Of these variables, nine are global variables, five are local variables, nine are linking variables, and there are two constraints. The problem in the paper is loosely coupled, meaning that IDF was an understandable choice for the problem architecture.

Only the hydrodynamics discipline utilises surrogate approaches in their multidisciplinary analyses, because the alternative, doing full computational fluid dynamics (CFD) analyses, would be prohibitively resource and time consuming. The optimization engine used for this problem is the DMOEOA optimizer, and the optimizer produces a valid Pareto front in around 300 generations.

Building envelope design

Yang et al [45] utilise a design variable minimisation approach in a similar way to that in [43], proposing design of experiments or sensitivity analysis approaches to identify the most influential design variables, although the authors acknowledge that advanced techniques are needed to ascertain relative importance of individual design variables. A two-subsystem problem is proposed with IDF and CO architectures. However, only preliminary results are shared, with the majority of the research occurring in pre-processing and case study quantification.

Launch vehicle design with Bayesian multi-objective optimization

In 2019, a launch vehicle optimization strategy was proposed using a variant of MO-MDO that used Kriging surrogate models to process the MDAs rather than the first principles equations [44]. This approach was chosen in the paper due to the complexity of the problem involved, as this can cause the problem to take a very long time to be optimized. Additionally, formulations for distributed (multi-level) and monolithic (single level) approaches are provided. In the single-level approach, coupled and decoupled formulations[1] are provided, with two disciplines accounting for the expendable and reusable launch vehicles. The reusable configuration contains four subdisciplines, while the expendable configuration contains two. In the multi-level design, a consistency variable is used in a similar way to that in [43], but the consistency functions at the disciplinary levels are slightly different.

2.3 Experimental choices

This section will outline some of the key choices made in the main body and research sections of the thesis, and explain why they have been chosen over other methods or benchmarks.

[1]where the decoupled formulation does not run multiple times to reach consensus, while the coupled formulation does

2.3.1 Use of the MDF and CO architectures

As explained in Section 2.1, there are many different architectures in the MDO literature, which are useful for different applications in different circumstances and contexts. MDF (a monolithic architecture) and CO (a distributed architecture) are some of the most thoroughly studied architectures in the MDO literature. Consequently, they have been selected for use in the research in this thesis for the following reasons:

- Their formulations, implementations, behaviours, advantages and drawbacks are well-known, meaning they are good starting points for developing multi-objective benchmarks in MDO.
- They have well-defined structures and are consistent throughout the literature.
- They include local, linking and global variables in their formulations. This is not always the case in MDO, as some formulations do not contain all types of variable, meaning a less generalisable problem. An example of this is in MDO of independent subspaces [109], where global variables do not exist; the only design variables are local.

2.3.2 Use of the ZDT test suite

The ZDT test suite [47], discussed in Section 2.2 is a very well-known MOP test suite and has been used extensively in the MOP research. However, it has been falling out of favour in the MOP community, with more other test suites such as BBOB-biobj [91, 92] and WFG [89] becoming more popular over time.

In this thesis, ZDT is used as a starting point for one of the key lines of research – establishing a new analytical MO-MDO test set. The reasons for this are:

- The ZDT test problems are separable. This is an important aspect because there are three different types of variables in MDO, in contrast to the one in non-MDO MOP. This needs to be represented in the MO-MDO benchmark problem. The separability permits the easy expression of these three types of variables, without changing the answer to the problem or the Pareto optimal front.

- The ZDT test problems have a known Pareto optimal front.

2.3.3 Use of the NSGA-II optimization engine

The number of multi-objective optimizers has increased massively since they were first conceptualised and developed. This is especially the case in bio-inspired optimizers – see [110] for a list of the many bio-inspired algorithms in the multi-objective optimization literature.

One of the best-known multi-objective optimizers is the Nondominated Sorting Genetic Algorithm 2 (NSGA-II) [111]. This optimizer was used for all optimization throughout the research chapters in this thesis. The reasoning for this is as follows:

- NSGA-II is a freely-available optimizer with multiple implementations accessible online, meaning it is easy to use in this thesis's research.
- NSGA-II has inbuilt constraint-handling capabilities.
- NSGA-II is fast to execute ($O(MN^2)$ where M is the number of objectives and N is the population size [111]).

2.4 Summary

In this chapter, key concepts in both MDO and MOPs have been presented, as well as prominent benchmarks in these fields of research, and a selection of applied problems. The concepts presented in Section 2.1 are important for the information contained in the rest of the thesis, from how MDO problems and architectures are structured, to the concept of Pareto domination. Some of the benchmarks are also important to know and understand, as they are important in later chapters.

In Chapter 3, the concepts and benchmarks presented in this chapter are used in a review of the key literature in the field of MO-MDO architectures, methods and strategies.

Chapter 3

Review of literature in MO-MDO

3.1 Introduction

3.1.1 Introduction to multidisciplinary design optimisation

The earliest research in MDO featured problems with only one objective to be minimised. Since then, a variety of MO-MDO approaches and problems have been developed.

However, the MO-MDO literature is fragmented, incomplete, lacks standardised variable naming conventions, and studies are often not set within the context of traditional MDO problems. This makes it difficult to align these studies with 'traditional' MDO, adding unnecessary complexity to the body of MO-MDO research as a whole.

In this literature review, the information in the existing literature is collated and the key research gaps and challenges are identified, providing recommendations and directions for future work in the area of MO-MDO. Extended design structure matrices (XDSMs) are used to illustrate the different approaches and architectures found in the existing MO-MDO literature. More information on XDSMs is given in Section 2.1.1.

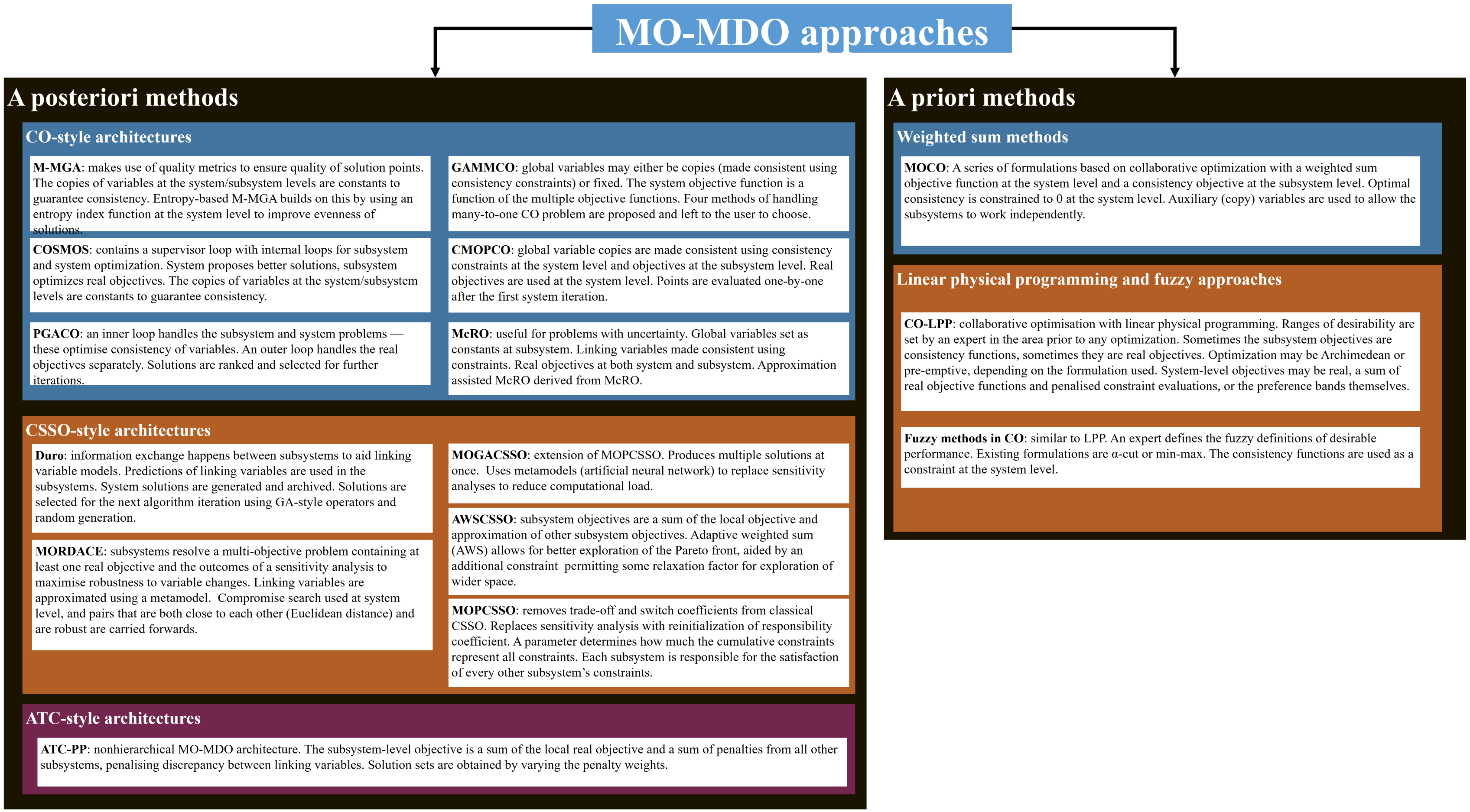
3.1.2 Contents of the review

This review is structured as follows. Section 3.1.3 presents a typology for a posteriori MO-MDO formulations, and uses this to evaluate the MO-MDO a posteriori methods. Section 3.2 covers some of the key a priori methods found in MO-MDO. Finally, Section 3.4 summarises the key points from the review.

A diagram of the main MO-MDO approaches reviewed in this paper is shown in Figure 3.1. These approaches are split into two main groups, as in Sections 3.2 and 3.3 (a priori methods and a posteriori methods). Of the a priori methods, there are two main 'groups' of architectures: the weighted sum methods and the linear physical programming and fuzzy method approaches. Of the a posteriori methods, there are three groups: the CO-style methods, the CSSO-style methods, and the ATC-style methods.

Note that in this literature review, the CSSO-style and ATC-style methods are discussed in Appendix B because they are not otherwise used in the rest of the thesis.

Figure 3.1: Overview of the MO-MDO approaches discussed in this review.



3.1.3 MO-MDO typology

For the purposes of defining a typology, here all MDO problems are generalised as having two levels: a system (upper) level and a subsystem (lower) level[1]. Using this generalisation, six possible types of multi-objective MDO structures can be defined based on the number of objectives in the problems at the subsystem and system levels. These objectives are either 0, where instead of an optimisation problem, some other analysis or co-ordination happens; 1, which is a single-objective problem; or >1, which is a multi-objective problem. Note that, for table entries where there is more than one objective in the subsystem, there needs only be one subsystem with a multi-objective problem.

Table 3.1 shows the different types of MO-MDO problems and which type they belong to. These are abbreviated using the letter T. To give an example, T5 is a type 5 problem. It has a system-level problem with more than one objective, and all its subsystem problems have only one objective.

Some entries in Table 3.1 are 'X' – these are architectures where there is no optimisation problem at all (top left), the MDO problem is both monolithic and single-objective (mid-left), or the problem is single-objective and CSSO-style (mid-top). Therefore, these are not included in the typology.

Some other important types are as follows:

- Type four is a multi-objective monolithic architecture, meaning there is an optimisation problem at the system level and the subsystem levels are analyses that are carried out by MDA modules. Examples of this type of problem can be found in [59, 58].
- Most CO-style MO-MDO architectures are either type two, three, five or six.

These definitions will be used throughout this subsection and in Table 3.2 to differentiate which architectures are equipped to handle the different multi-objective CO problems.

[1] In some architectures, such as some forms of Analytical Target Cascading (ATC) [70, 112] there are more than two levels, or the formulation is non-hierarchical, but for the purposes of this paper, these will be left out of the typology and identified when necessary.

		Subsystem: number of objectives		
		0	1	>1
System: number of objectives	0	X	X	T1
	1	X	T2	T3
	>1	T4	T5	T6

Table 3.1: A table showing the six types of MO-MDO systems.

3.2 A priori methods

In the MOP literature, there are three common ways that a decision maker (DM) can express preferences for optimising a system or engineered product. These are a priori, a posteriori, and interactive methods.

In a priori optimsiation methods, the DM has to express preferences before the optimisation process is carried out. A priori methods are often expressed using mathematical weights indicating preference for different objectives, or using preference bands, which are similar to the fuzzy sets proposed by Zadeh [113]. A priori methods can be useful when the DM already understands how their choices affect the system, they have in-depth knowledge of the system, or they already know how they want the system to behave following optimisation. It is also helpful if the DM isn't available to engage regularly with the optimisation process as in interactive methods, or won't be available for solution assessment in the a posteriori methods. The use of a priori methods should be dependent on the situation – by necessity they omit alternative designs which in some cases may be considered a drawback; however, if this is the case, interactive or a posteriori methods should be sought.

A priori optimisation is the most obvious method for handling MO-MDO problems, because they are functionally very similar to single-objective MDO, meaning it is necessary to make only minimal alterations to these MDO architectures and strategies.

Haimes et al. published a review of multi-objective MDO methods in 1988 [114], covering weighted sum and linear physical programming, among other approaches.

In addition, an early version of MOP trade-offs in MDO were discussed, although this work was produced before the advent of genetic algorithms and evolutionary computation methods in the late 1990s. The benefits of decomposing a system into multiple subsystems (referred to as a 'subordinate' in an 'organisation') are noted well as being able to decrease the complexity of the problem and reflecting hierarchical engineering organisations.

3.2.1 Weighted sum methods

In weighted sum methods, system objectives are weighted numerically and then summed together to create a composite objective which is then used for optimisation. Weighted sum objective function approaches are the most straightforward and obvious approach to multi-objective optimisation in MDO, because it requires no changes are made to the architecture of the problem, and the weighted sum of objective functions can essentially be treated as one objective function. However, there are two difficulties with this method. Firstly, choosing the weights for the objectives appropriately can be tricky, and there is no set method for carrying this out 'correctly'. Secondly, if the trade-off surface is non-convex, it is impossible to find all Pareto optimal solutions (see [115, 116, 117]).

The weighted sum method has been applied to analytical target cascading (ATC) [118] (see Appendix B) and collaborative optimisation (CO) [119, 120, 57].

Multi-objective Collaborative Optimisation (MOCO)

Tappeta and Renaud proposed three potential approaches for multi-objective problems within a CO architecture, named multi-objective collaborative optimisation (MOCO) [120]. All of these make use of weighted sum objective functions at the system level, with a sum of square differences between variables and target variables comprising a consistency function at the subsystem level, meaning that the formulation follows a CO-style architecture. This approach is a priori because the designer of the system needs to choose the weights that are applied to the objective functions at the system level. As with all a priori formulations, the inherent difficulty in this originates with the need of the designer to fully understand what they need to obtain from the system, rather than being able to pick a solution at the end of the optimisation process.

Each formulation contains three subsystems. In the first formulation, the system-level optimiser independently invokes the subsystem analyses, and the objective function evaluations for each subproblem are returned to the system-level optimiser. However, this formulation can lead to a very large problem at the system level, and the consistency constraints applied the local variables can lead to highly restricted solution sets. Also, as mentioned in [24], the equality constraint $J^* = 0$ at the system level can cause convergence problems.

There are three key differences between the 'traditional' CO architecture and the MOCO architectures proposed in [120], disregarding the weighted sum objective at the system level; firstly, auxiliary variables $\mathbf{u_i}$ are used in place of the incoming copies of linking variables from other disciplines to allow disciplines to execute concurrently. Secondly, the disciplines return the sum of squared errors between 'active' and copy variables to the system optimizer, but not the variables themselves. Thirdly, the Tappeta and Renaud formulations have the system-level consistency constraint $d_i^* = 0 - d_i^*$ is returned from the subsystems as a constraint. In classic CO, the system-level problem has its own consistency constraint, independent of the subsystem-level objectives.

Each formulation has additional differences between itself and the classic CO formulation, but these will be detailed in the architecture summaries below. Note that, for these, a hat ˆ is used to show a copy variable.

The first formulation (system-level problem) is an extension of [121]. The system optimizer sends copies of the global, local and auxiliary variables to the subsystem optimizers. The active design variables used at the subsystem are the global, auxiliary, and local variables for the subsystem k. A subspace analysis (also referred to as an MDA in comparative MDO literature) is used to calculate the linking variables for subsystem k. The sum of squared differences between the linking variables and the auxiliary variable k is used in the subsystem objective.

The subsystem optimizer is permitted to choose the active variables corresponding to the copy auxiliary variables sent by the system optimizer, contingent on the output of the subspace analyses.

The objective functions at the system level are obtained via the subspace analyses independently, using the copy variables chosen by the system optimizer, and then returned to calculate the weighted sum objective function.

$$\begin{aligned} \min \quad & F(\hat{\mathbf{x}}) = \sum_{i=1}^{2} w_i f_i(\hat{\mathbf{x}}) \\ \text{s.t.} \quad & J_i^* = 0, \quad i = 1,2,3 \\ & \hat{\mathbf{x}}_{min} \leq \hat{\mathbf{x}} \leq \hat{\mathbf{x}}_{max} \\ \text{where} \quad & \hat{\mathbf{x}} = [\hat{z}, \hat{x}_i, \hat{u}], \qquad i = 1,2 \end{aligned} \tag{3.1}$$

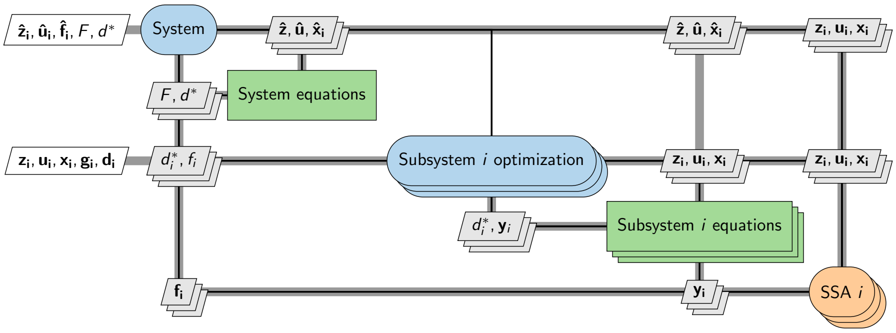
and the subsystem i problem is

$$\begin{aligned} \min \quad & J_i(z_i, u_i, x_i) = (z_i - \hat{z}_i)^2 + (x_i - \hat{x}_i)^2 + \sum_{j=2}^{3}(u_{ji} - \hat{u}_{ji})^2 + \sum_{j=2}^{3}(y_{ij} - \hat{u}_{ji})^2 \\ \text{s.t.} \quad & g_i(z_i, u_i, x_i) \geq 0 \\ & z_{i_{min}}, \leq z_i \leq z_{i_{max}} \\ & u_{i_{min}} \leq u_i \leq u_{i_{max}} \\ & x_{i_{min}} \leq x_i \leq x_{i_{max}} \end{aligned} \tag{3.2}$$

The XDSM for the first Tappeta and Renaud MOCO formulation is shown in Figure 3.2.

In the second formulation, based on the method presented in [122], the target and evaluated values of the objective function for each subsystem are used in the consistency equations. This means that the problem cannot be multi-objective at the subsystem level. The difference between the target and the real value of the subsystem objective function replaces the difference between the real and target local variables as in formulation 1, meaning that the subsystems have the freedom to change the local design variables as needed. The second formulation's system-level

Figure 3.2: The XDSM for Tappeta and Renaud's first a priori MOCO formulation.



problem is

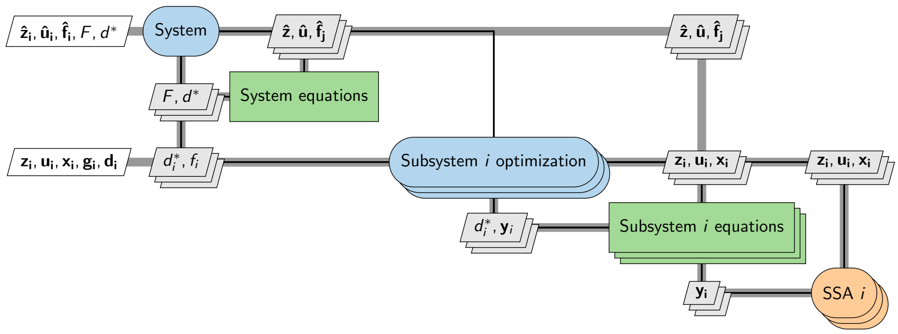
$$
\begin{aligned}
\min \quad & F = \sum_{i=1}^{2} w_i \hat{f}_i \\
\text{s.t.} \quad & J_i^* = 0, \quad i = 1, 2, 3 \\
& \hat{\mathbf{x}}^{min} \leq \hat{\mathbf{x}} \leq \hat{\mathbf{x}}^{max} \\
\text{w.r.t} \quad & \hat{\mathbf{x}}, \hat{f}_1, \hat{f}_2
\end{aligned}
\tag{3.3}
$$

and the subsystem-level problem is

$$
\begin{aligned}
\min \quad & J_k(\mathbf{x_{ssk}}) = (\mathbf{z}_k - \hat{\mathbf{z}}_k)^2 + (f_1 - \hat{f}_1)^2 + \sum_{i=2}^{3} (\mathbf{u}_{ik} - \hat{\mathbf{u}}_{ik})^2 + \sum_{i=2}^{3} (\mathbf{y}_{ki} - \hat{\mathbf{u}}_{k1})^2 \\
\text{s.t.} \quad & g_k(\mathbf{x_{ssk}}) \geq 0 \\
& \mathbf{x}_{ssk}^{min} \leq \mathbf{x}_{ssk} \leq \mathbf{x}_{ssk}^{max} \\
\text{where} \quad & x_{ssk} = [z_i, u_{ik}, x_k]
\end{aligned}
\tag{3.4}
$$

In the third formulation, neither the local variables nor the objective functions are used to evaluate the consistency function at the subsystem level. This can result in a problem with a smaller dimensionality than either of the other two formulations if the problem is sparse. The third formulation's system-level problem is

Figure 3.3: The XDSM for Tappeta and Renaud's second a priori MOCO formulation.



$$
\begin{aligned}
\min \quad & F = \sum_{i=1}^{2} w_i f_i \\
\text{s.t.} \quad & J_i^* = 0, \quad i = 1, 2, 3 \\
& \hat{\mathbf{x}}^{min} \leq \hat{\mathbf{x}} \leq \hat{\mathbf{x}}^{max} \\
\text{w.r.t} \quad & \hat{\mathbf{x}}
\end{aligned}
\tag{3.5}
$$

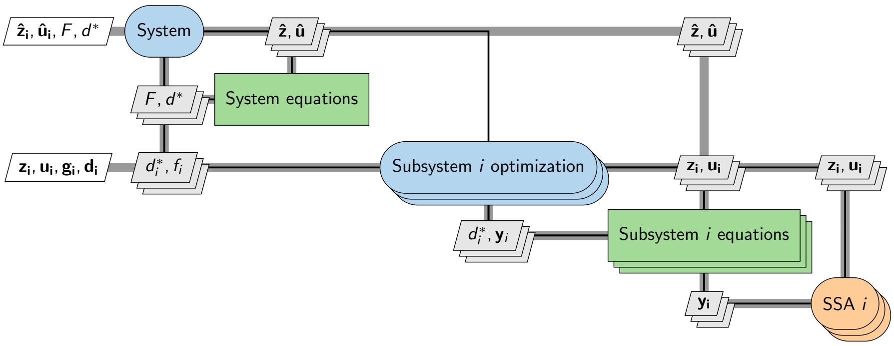
and the subsystem-level problem is

$$
\begin{aligned}
\min \quad & J_k(\mathbf{x_{ssk}}) = (z_k - \hat{z}_k)^2 + \sum_{i=2}^{3} (u_{ik} - \hat{u}_{ik})^2 + \sum_{i=2}^{3} (y_{1i} - \hat{u}_{i1})^2 \\
\text{s.t.} \quad & g_1 \geq 0 \\
& \mathbf{x}_{ssk}^{min} \leq \mathbf{x}_{ssk} \leq \mathbf{x}_{ssk}^{max} \\
\text{w.r.t} \quad & \mathbf{x}_{ssk} = [z_k, u_{ik}, x_i]
\end{aligned}
\tag{3.6}
$$

Note that none of these formulations are scalable in their current form because the formulation is set up explicitly for problems with three subsystems and two objectives.

A critique of MOCO was presented in [123] with four main points: the determination of step size is undetermined, especially when there are conflicts between subsystems; the use of sequential quadratic programming in MOCO means that the optimiser is likely to terminate at local minima; the use of a weighted sum objective

Figure 3.4: The XDSM for Tappeta and Renaud's third a priori MOCO formulation.



function means that it is difficult to obtain a large diversity of solutions; and the delinquency of the MOCO formation also described in [24].

Improved collaborative optimisation

Improved collaborative optimisation was proposed in 2010 in [119]. It is noted by the authors that many complex optimisation problems encountered in industry have more than one objective to be minimised at once, and that weighted sum approaches are a possible solution to that, given that most of the CO literature at the time was single-objective. The formulation proposed here is similar in structure to MOCO formulation one, proposed by Tappeta and Renaud in [120]; the subspace analyses are invoked both following subsystem optimisation, and then again following system optimisation using the DVs from the system problem, returning the subsystem objective functions for use in the composite weighted objective function. This is compared with the objective functions obtained using the design variables from the individual subsystem optimisations. There are two main difference between these two formulations. Firstly, improved collaborative optimisation normalises the objective function terms within the composite objective function using the returned values from the subsystem optimisations, F_i^*. Secondly, improved collaborative optimisation introduces a relaxation on the consistency constraint at the system level, while the consistency constraint in MOCO is an equality constraint.

The system-level problem is

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} w_i \left| \frac{F_i(\mathbf{x})}{F_i^*} \right| \\ \text{s.t} \quad & J_i^*(\mathbf{x}) \leq \epsilon, \qquad i = 1, ..., N \end{aligned} \tag{3.7}$$

and the subsystem-level problems are expressed as

$$\begin{aligned} \min \quad & F_i^* + J_i^*(\mathbf{x}) \\ \text{s.t} \quad & g_i(\mathbf{x}), \qquad i = 1, ..., N \end{aligned} \tag{3.8}$$

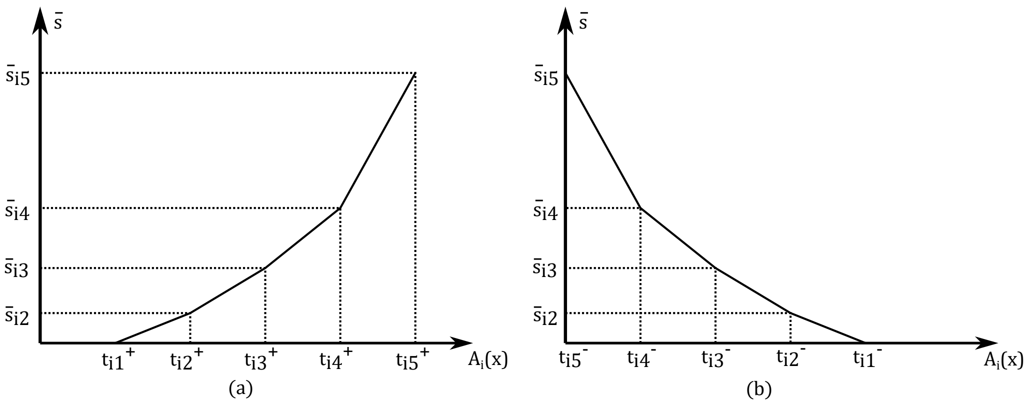
3.2.2 Linear physical programming in MDO

Linear physical programming (LPP) is an a priori method that allows the system designer to set meaningful preferences for design outcomes. Its application to multi-objective MDO problems was in response to the problems posed by the MOCO formulation presented in Section 3.2.1.

These preferences are usually set as preference bands (referred to as ranges of desirability), ranging from unacceptable designs, where the design's poor performance according to an aggregate objective function is intolerable, to ideal designs that satisfy all system requirements. These ranges of desirability are labelled as 1-4S or 1-4H, where S represents soft aggregate functions that seek to minimise (1S) or maximise (2S) as far as possible, obtain the best possible value (3S) or obtain the largest possible range (4S); hard aggregate functions aim to obtain values beneath (1H) or above (2H) a prescribed value, obtain an aggregate function value equal to a given value (3H), or obtain a result within a given range (4H). The soft aggregate functions are analogous to objectives, while the hard aggregate functions are analogous to constraints. These preferences are assigned outside of the optimisation problem, then integrated as constraints at both the system and the subsystem levels. See Messac's work on LPP [3] and Ilgin and Gupta's review on physical programming [124] for more details.

These classes can be visualised using the plot of the class function regions, replicated in Figure 3.5. The value of the class function is $\bar{s}$. Smaller values of $\bar{s}$ are considered better [2, 125]. The design achievements are represented using t_{ij}, where i indicates the physical objective number and j is the quality of the solution (again, smaller is better). The superscripts on the achievement indicate whether the goals are over-achieved (i.e. they overshoot the target) or under-achieved (they undershoot the target). The deviations between these and the physical objectives f_i make up the optimisation problem at the subsystem level.

Figure 3.5: Class functions 1S - the minimisation problem (a) and 2S - the maximisation problem (b). Class functions 3S and 4S can be constructed from a combination of 1S and 2S so they are not included in this diagram. Diagram adapted from [1, 2, 3].



Physical programming is useful in problems where desirable objective boundaries aren't clearly defined, or need significant expert input. For example, physical programming might be used in social systems where optimal outcomes are poorly defined. Additionally, the architecture responds strongly to uncertainty in design variables [125].

MDO LPP methods have been used in the design of a rolling mill stand [2], a speed reducer [5], race car design [1], and a two-bar structure problem [6].

CO-LPP

McAllister et al. [1, 4] used linear physical programming to define preferences in a MDO problem with the collaborative optimisation formulation using the design of a race car as an example. Linear physical programming is integrated into the CO architecture using a compromise decision support problem [126].

Deviation variables d_i are calculated as the difference between the actual attained value from the optimisation A_i, also known as the attainment function, and the goal range assigned by the designer $t_{i,k}$.

$$d_{i,k} = t_{i,k} - A_{i,k}(x) \tag{3.9}$$

The deviation variables can be split into the negative and positive deviations d_i^- and d_i^+ respectively, $d_i = d_i^- - d_i^+$. Therefore,

$$A_{i,k}(x) - d_{i,k}^+ + d_{i,k}^- = t_{i,k} \tag{3.10}$$

which is used as a constraint in the system-level problem. The supporting statements

$$\begin{aligned} d_{i,k}^+ \cdot d_{i,k}^- &= 0 \\ d_{i,k}^+, d_{i,k}^- &\geq 0 \end{aligned} \tag{3.11}$$

must also hold, and are therefore also considered constraints in the problem. More information can be found in [127, 126].

Their CO system-level formulation modified to include the results of the preference bands is as follows:

$$
\begin{aligned}
\min \quad & F = [f_1, f_2, f_3, f_4] \\
\text{s.t.} \quad & \mathbf{g_{sys}}(\hat{\mathbf{x}}, \mathbf{z}) \leq 0 \\
& \mathbf{z} - \hat{\mathbf{z}} = 0 \\
& A^{+}_{i21}(\hat{\mathbf{x}}, \mathbf{z}) + d^{-}_{i21,k} - d^{+}_{i21,k} = t^{+}_{i21,k}, \quad i_{21} = 1, ..., n_{\bar{s}1}, k = 1, ..., 4 \\
& A^{-}_{i22}(\hat{\mathbf{x}}, \mathbf{z}) + d^{-}_{i22,k} - d^{+}_{i22,k} = t^{+}_{i22,k}, \quad i_{22} = 1, ..., n_{\bar{s}2}, k = 1, ..., 4 \\
& d^{-}_{i22,k}, d^{+}_{i21,k} \geq 0 \\
& d^{-}_{i21,k} \cdot d^{+}_{i21,k} = 0 \\
& d^{-}_{i22,k} \cdot d^{+}_{i22,k} = 0 \\
& x_{i1_{min}} \leq x_{i1} \leq x_{i1_{max}}, \quad i_1 = 1, ..., n_z + n_x + n_y \\
\text{where} \quad & f_j = \sum_{i=1}^{n_{\bar{s}1}+n_{\bar{s}2}} w_{i,5-j}(d^{+}_{i,5-j} + d^{-}_{i,5-j})
\end{aligned}
\tag{3.12}
$$

Where $n_{\bar{s}1}$ is the number of 1S goals, and $n_{\bar{s}2}$ is the number of 2S goals. g_{sys} are the inequality constraints at the system level. k is the number of class functions and n_z, n_x and n_y are the total number of global, local and linking variables, respectively.

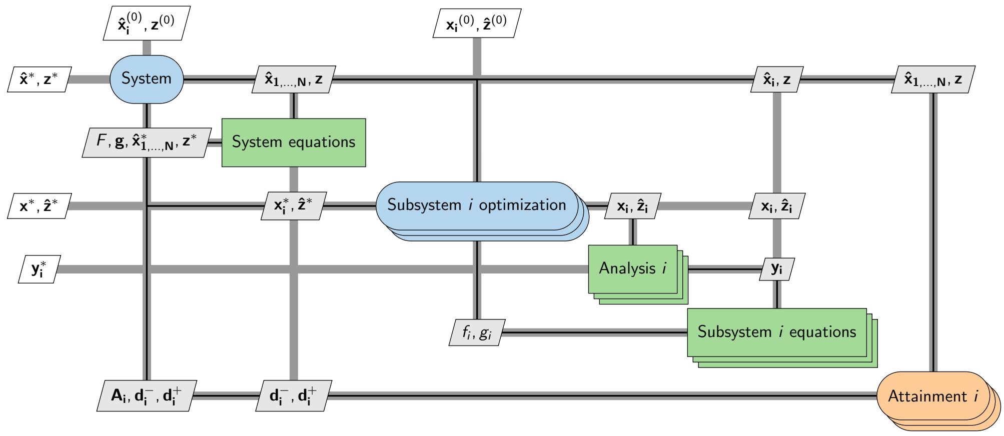
Note the objective function $F = [f_1, f_2, f_3, f_4]$. As in [126], the lexicographic minimum is the objective. This means that the lower indices are preferred over higher indices (meaning also that the weighted deviation for class 1 is infinitely more important than the weighted deviations for lower classes). The objective function formulation is known as pre-emptive in the literature [3, 126].

The subsystem-level formulation is

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n_{x_i}} (\hat{\mathbf{x}}_{\mathbf{i}} - \mathbf{x}_i)^2 + \sum_{i=1}^{n_z} (\mathbf{z}_i - \hat{\mathbf{z}}_{\mathbf{i}})^2 \\
& f(d^{+}_{i}, d^{-}_{i}) \\
\text{s.t.} \quad & \mathbf{g}_{sub_p}(\mathbf{x}, \hat{\mathbf{z}}) \leq 0 \\
& x_{i1_{min}} \leq x_{i1} \leq x_{i1_{max}}
\end{aligned}
\tag{3.13}
$$

Where $\mathbf{g}_{sub_p}$ are the constraints for the subsystem-level problem and $f(d_i^+, d_i^-)$ are the local objectives in the subsystem problem. The XDSM of McAllister et al.'s CO-LPP approach is shown in Figure 3.6.

Figure 3.6: The XDSM diagram of CO-LPP adapted from [1, 4].



In Li et al.'s article [5] investigating multi-objective LPP in multidisciplinary problems, there are a few main differences. Firstly, the design preferences are implemented at the subsystem level rather than the system level. Secondly, the Archimedean formulation is used for the objective function instead of the preemptive formulation, which is used in McAllister et al.'s CO-LPP.

The system-level problem is given by

$$\begin{aligned} \min \quad & F(z) \\ \text{s.t.} \quad & J_i^*(z) = \sum_{j=1}^{s_i} (z_j - \hat{z}_{ij})^2 \leq \varepsilon, \quad i = 1, ..., n \\ & J_{n+1}^*(z) = \sum_{i=1}^{n} J_i^*(z) \leq E \end{aligned} \tag{3.14}$$

Where E is directly proportional to the inconsistency $I = \frac{\sum_{i=1}^{n} \sqrt{J_i(x_i)}}{n}$ between the subsystems. It can be seen that the system-level constraints involve reducing the inconsistency between subsystems as much as possible. The subsystem-level formulation is slightly modified to contain weights for both the positive deviation and the negative deviation. Designer preferences are incorporated at the subsystem level, leaving the system-level optimiser to act as a coordinator, which is more in line with traditional collaborative optimisation than the formulation used by McAllister et al. above.

$$
\begin{aligned}
\min \quad & \tilde{J}_i = \sum_{p=1}^{n_s}\sum_{s=2}^{5}(\tilde{w}_{ps}^{-}d_{ps}^{-} + \tilde{w}_{ps}^{+}d_{ps}^{+}) \\
\text{s.t.} \quad & c_i(x_i) \leq 0 \\
& A_{ip}(x_i) + d_{ps}^{-} \geq t_{p(s-1)}^{-}, \quad A_{ip}(x_i) \geq t_{p5}^{-}, \quad A_{ip}(x_i) \geq t_{p5}^{-}, \quad p = 1,2,4 \\
& A_{ip}(x_i) + d_{ps}^{+} \leq t_{p(s-1)}^{+}, \quad A_{ip}(x_i) \leq t_{p5}^{+}, \quad A_{ip}(x_i) \leq t_{p5}^{+}, \quad p = 2,3,4 \\
& \sum_{j=1}^{s_i}(z_{ij} - \hat{z}_j)^2 - max\left\{\sum_{j=1}^{s_i}(z_{ij} - \hat{z}_j)^2,\ k_d d_0\left(1 - \frac{SS_i}{SS_n}\right)\right\} \leq 0
\end{aligned}
\tag{3.15}
$$

where p represents the number of physical objectives and s is the function class number. The final constraint contains a term $k_d d_0\left(1 - \frac{SS_i}{SS_n}\right)$ which is known as the *dynamic intermediate parameter* which decreases as the number of supervisor iterations increases. k_d is a parameter that is chosen by the decision maker and d_0 is the distance between the optimal value of the unconstrained problem and the nearest point within the feasible region of the constrained problem. SS_i is the current supervisor iteration and SS_n is the projected total number of supervisor iterations. This means that, at the final supervisor iteration, the final constraint evaluation becomes 0.

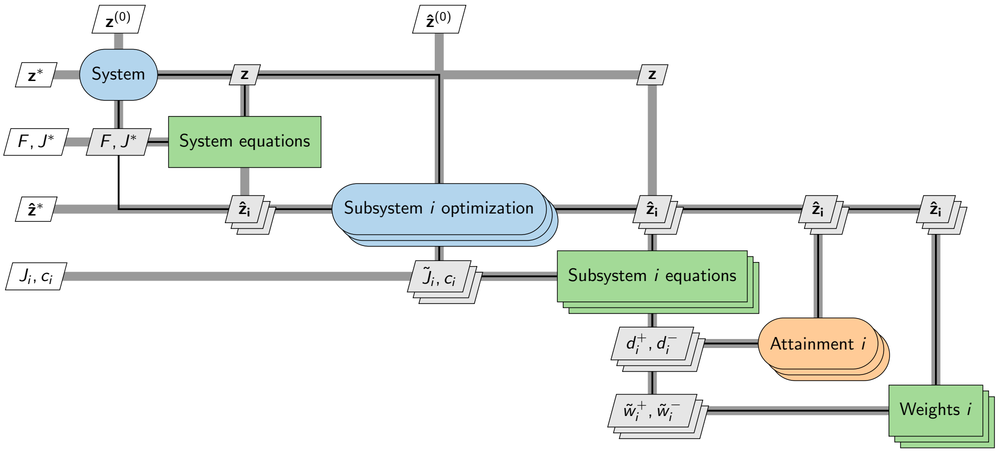
Note that the first five constraints mirror the statements supporting the integration of LPP and CO found in Equations 3.9 to 3.11 [127, 126].

The XDSM of this method is shown in Figure 3.7.

In their later paper, Li et al. add a dynamic weight γ to the objective function in this formulation [2],

$$\tilde{J}_i = \sum_{p=1}^{n_s}\sum_{s=2}^{5}(\tilde{w}_{ps}^{-}d_{ps}^{-} + \tilde{w}_{ps}^{+}d_{ps}^{+}) + \gamma(J_i(x_i)) \tag{3.16}$$

Figure 3.7: The XDSM diagram of CO-LPP adapted from [5].



which penalises inconsistency between the subsystems.

A similar method is presented by Baril et al. [6] using linear physical programming to incorporate preferences named the collaborative interactive multi-objective (CIMO) algorithm. The system-level optimiser minimises each of the physical objectives and re-writes the constraints as weighted penalty terms:

$$\begin{aligned} \min \quad & \sum_{p=1}^{m} \left(F_p(z) + \gamma \sum_{j=1}^{J} g_j(\hat{x}_j, z) \right) \\ \text{s.t.} \quad & z_{min} \leq z \leq z_{max} \end{aligned} \tag{3.17}$$

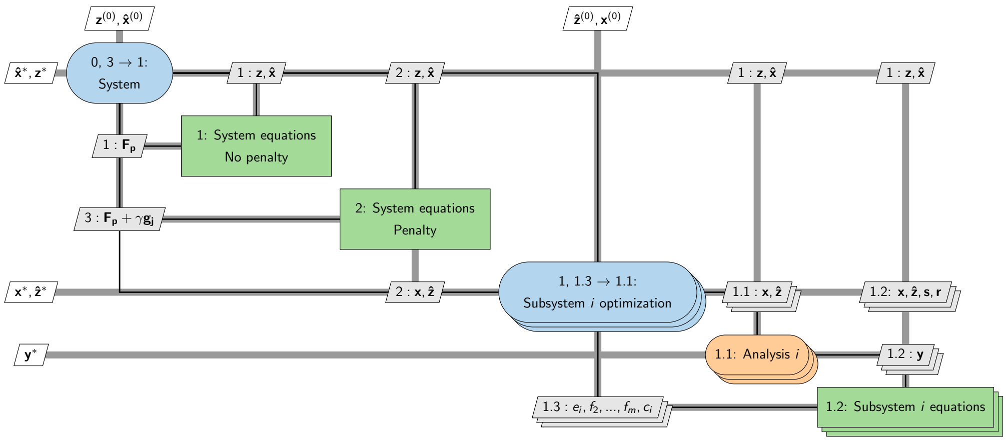
Preferences are expressed at the subsystem level using the Archimedean objective function formulation. Penalty functions are utilised on consistency constraint violations to widen the potential pool of solutions. The subsystem i formulation is

$$\begin{aligned} \min \quad & \{e_i(s,r), f_2(x,\hat{z}), ..., f_m(x,\hat{z})\} \\ \text{s.t.} \quad & x + s - r = z, \quad y + s - r = z \\ & x_{min} \leq x \leq x_{max} \\ & \hat{z}_{min} \leq \hat{z} \leq \hat{z}_{max} \end{aligned} \tag{3.18}$$

where s are slack variables and r are surplus variables.

Most importantly, the CIMO approach acknowledges and uses linking variables in its formulation, unlike the CO-LPP formulations discussed above.

Figure 3.8: The XDSM diagram of CIMO adapted from [6].



In short, McAllister et al. implement the decision maker's preferences expressed via LPP methods in the system-level problem, while Li et al. and Baril et al. implement the same information at the subsystem level. The effect this has is that the first formulation is constrained by the linkages between the system — inter-subsystem relationships aren't affected by the decision maker's preferences, leaving the system-level optimiser to both co-ordinate the subsystem responses for an optimal solution and to implement the preferences. The second and third formulations listed allow the subsystems to handle the preferences, and leaves the system-level optimiser to co-ordinate only the subsystem responses. This approach is closer to the non-LPP collaborative optimisation method, and can reduce pressure on the system-level optimiser, especially in cases where there are many subsystems. However, preferences for the system-level objective function can be hard to articulate and implement.

3.2.3 Fuzzy methods in CO

Fuzzy mathematics have also been used in a priori multi-objective MDO. Fuzzy methods are similar to physical programming and linear physical programming in that ranges of targets are set, which allows specialists in the relevant area of design to specify ranges of acceptability a priori.

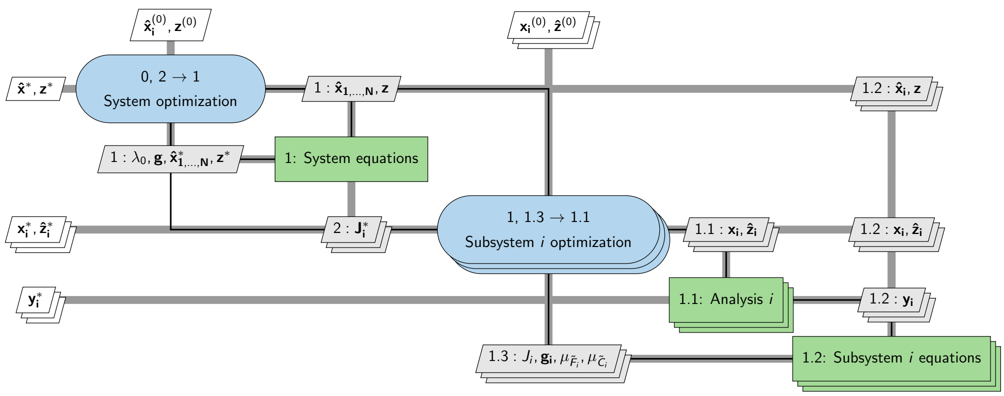
Huang et al. [128] developed a method building on CO using fuzzy satisfaction and sufficiency degree models – called the Fuzzy Satisfaction Sufficiency Degree Collaborative Optimisation (FSSDCO). The fuzzy satisfaction degree is a metric that quantifies the degree to which the objectives are satisfied by a solution, the definitions of which are set by a decision-maker. The fuzzy sufficiency degree is similar to the fuzzy satisfaction degree, but it quantifies the degree to which the constraints are satisfied by a solution. Both of these are evaluated in the subspace analysis.

The multiple objectives at the system level were handled using two approaches: the α-cut method and the max-min method. In the max-min method, the objective at the system level is to maximise the *acceptable degree level* $\mu_{\Phi_i}(\mathbf{x})$, which is the smallest value between the fuzzy satisfaction and fuzzy sufficiency degrees.

$$\mu_{\Phi_i}(\mathbf{x}) = \left(\wedge_{i=1}^{n_o} \mu_{\tilde{F}_i(\mathbf{x})}\right) \wedge \left(\wedge_{i=1}^{n_c} \mu_{\tilde{C}_i(\mathbf{x})}\right) \tag{3.19}$$

where n_o is the number of objectives, n_c is the number of constraints, $\mu_{\tilde{F}_i(\mathbf{x})}$ is the fuzzy satisfaction degree for objective i, and $\mu_{\tilde{C}_i(\mathbf{x})}$ is the fuzzy sufficiency degree for constraint number i. The wedge operator $\wedge$ is the Min, as established in [129].

Figure 3.9: The XDSM diagram of max-min FSSDCO adapted from [7].



At the system level, the objective is to maximise the acceptable degree level

λ_0.

$$
\begin{aligned}
\max \quad & \lambda_0 \\
\text{w.r.t} \quad & \mathbf{z}, \hat{\mathbf{x}}, \hat{\mathbf{y}} \\
\text{s.t.} \quad & J_i^* = 0, \quad i = 1, ..., N
\end{aligned}
\tag{3.20}
$$

where N is the total number of disciplines and J_i^* is the consistency value obtained at the subsystem levels, as in classic CO.

In the optimisation problem at the subsystem level, the objective is to maximise $\mu_{\Phi_i}(\mathbf{x})$:

$$
\begin{aligned}
\max \quad & \mu_{\Phi_i}(\mathbf{x}) = \lambda_i, \quad i = 1, ..., N \\
\text{s.t.} \quad & \mu_{\tilde{C}_{i,j}}(\mathbf{x}) \leq \lambda_i \omega_{i,j}, \quad j = 1, ..., n_{c_i} \\
& \mu_{\tilde{F}_{i,k}}(\mathbf{x}) \leq \lambda_i \omega_{i,k}, \quad k = 1, ..., n_o \\
& 0 \leq \lambda_i \leq 1 \\
& 0 \leq \omega_i \leq 1
\end{aligned}
\tag{3.21}
$$

where ω is a weight that emphasises the satisfaction and sufficiency degrees. The XDSM for the max-min FSSDCO method is shown in Figure 3.9.

In the α-cut method, only the minimum of the fuzzy satisfaction degrees (relating to the objective function) is maximised at the system level. This subsystem-level problem is expressed as

$$
\begin{aligned}
\max \quad & \wedge_{i=1}^{m} \mu_{\tilde{F}_i}^0 \\
\text{w.r.t} \quad & \mathbf{z}, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \mu_{\tilde{F}_i}^0 \\
\text{s.t.} \quad & J_i^* = 0, \quad i = 1, ..., N.
\end{aligned}
\tag{3.22}
$$

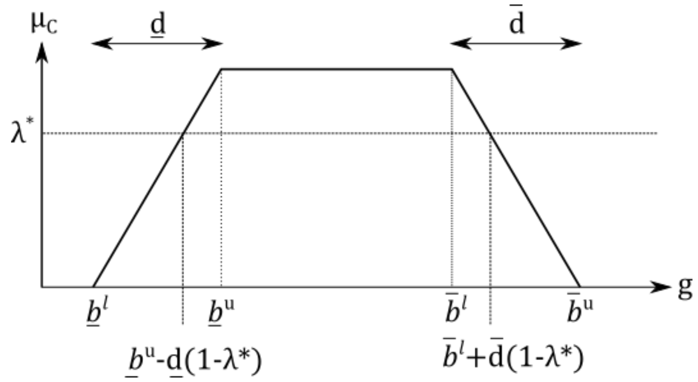
At the subsystem level, the consistency function is minimised, as in classic CO. The MDAs take on the variables $\hat{\mathbf{z}}, \mathbf{x_i}$ and return the linking variables $\mathbf{y}_i$ and the fuzzy satisfaction degree $\mu_{\tilde{F}_i}^0$. In addition to these components, an additional subsystem optimum level analysis is used along with the MDAs, and finds some optimal cut set denoted λ_i^* for subsystem i. In the XDSM showing the α-cut method in Figure 3.11, the subsystem optimum level analysis is not shown as having any inputs from the rest of the optimisation process – this is because it receives no variables from the subsystem optimiser, and depending on the method chosen to find λ_i^*, it may rely on having experts formulate and evaluate the value [130].

The subsystem-level problem is

$$\begin{aligned} \max \quad & d_i = (\hat{\mathbf{z}}_\mathbf{i} - \mathbf{z})^2 + \sum_{j=1}^{n_{x_i}} (\mathbf{x_j} - \hat{\mathbf{x}}_\mathbf{j})^2 + \sum_{j=1}^{n_{y_i}} (\mathbf{y_j} - \hat{\mathbf{y}}_\mathbf{j})^2 + \sqrt{(\mu_{\tilde{F}_i} - \mu^*_{\tilde{F}_i})} \\ \text{w.r.t} \quad & \mathbf{z}, \hat{\mathbf{x}}, \hat{\mathbf{y}} \\ \text{s.t.} \quad & g_j(\mathbf{z}, \hat{\mathbf{x}}, \hat{\mathbf{y}}) \leq \bar{b}_j^l + \bar{d}_j(1 - \lambda_i^*), \quad j = 1, ..., n_{c_l} \\ & g_j(\mathbf{z}, \hat{\mathbf{x}}, \hat{\mathbf{y}}) \geq \underline{b}_j^u - \underline{d}_j(1 - \lambda_i^*), \quad j = n_{c_u} + 1, ..., n_{c_l} \end{aligned} \tag{3.23}$$

where n_{c_u} is the number of maximum-value constraints (i.e. constraining the maximum value) and n_{c_l} is the number of minimum-value constraints. For a minimum-value constraint, $\underline{b}^u$ is the upper limit of the fuzzy constraint set and $\underline{b}^l$ is the lower limit of the set (e.g. where $\mu_{\tilde{C}}(\underline{b}^u) = 1$ and where $\mu_{\tilde{C}}(\underline{b}^l) = 0$). For a maximum-value constraint, the statement is similar, but $\mu_{\tilde{C}}(\bar{b}^u) = 0$ and $\mu_{\tilde{C}}(\bar{b}^l) = 1$. An illustration for a fuzzy set with linear edges is provided in Figure 3.10.

Figure 3.10: An example of a fuzzy constraint set adapted from [7].

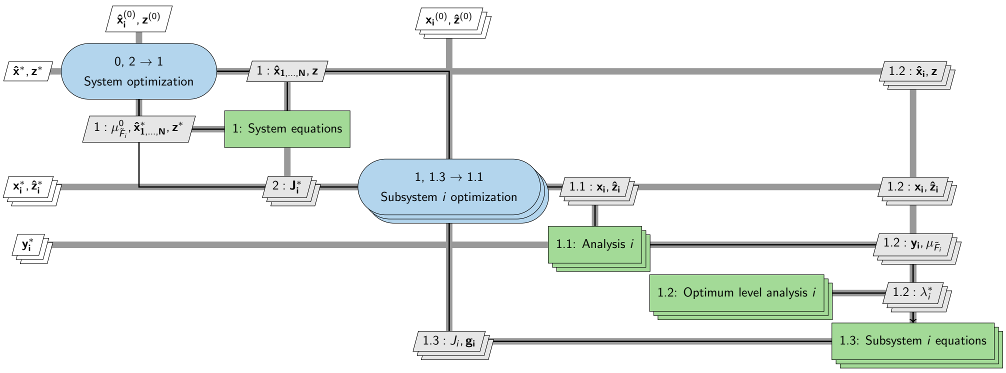


The XDSM for the α-cut FSSDCO method is shown in Figure 3.11.

3.3 A posteriori methods

In a posteriori methods, a set of solutions is produced by the optimiser (as opposed to a priori optimisation where only one solution is produced) and the DM expresses

Figure 3.11: The XDSM diagram of the α-cut FSSDCO method adapted from [7].



their preferences by selecting one solution from the set. A posteriori MO-MDO requires a more involved approach at the MDO architecture level because there isn't always an obvious strategy to propagating and transferring multiple solutions between the system and the subsystem levels, especially in distributed architectures such as CO.

The existing literature on a posteriori MO-MDO methods is mostly focused on CSSO- and CO-style architectures, with a few using the ATC architecture. These are documented in Subsections B.2.1, B.2.2 (both in Appendix B) and 3.3.1.

3.3.1 CO-style MO methods

Architectures with features of genetic algorithms

Some of the a posteriori multi-objective CO architectures build features or steps of genetic algorithms into their structure. For example, this may mean incorporating a ranking and selection process to prevent a pool of solutions from growing too large while choosing the best solutions to go forward (for example in [11]), or using mutation or crossover strategies to better explore all areas of the design space. In addition, many of the architectures incorporating genetic algorithm features use genetic algorithms to solve the optimization problems at the system or subsystem levels. Two architectures with features of genetic algorithms are included here.

Pareto genetic algorithm-based collaborative optimization (PGACO)

PGACO [8] was developed in response to the limitations of Tappeta and Renaud's MOCO [120] by making several changes, but especially the incorporation of a genetic algorithm in the architecture. This has the following effects

1. It reduces the likelihood that the results outputted from the architecture fall into local minima and stay there;
2. The architecture generates a set of potential solutions, which are missing with weighted-sum a priori methods;
3. In single-objective CO, the Karush-Kuhn-Tucker criteria are not met (see also [24, 8], which can be a problem for termination in some optimizers. Pareto-based GAs do not suffer from this.

In the PGACO architecture, there are essentially two problems, solved with an inner loop and an outer loop.

The inner loop solves a consistency problem with the system constraints at the subsystem. In the outer loop, a set of physically meaningful objective functions $F_{1,\ldots,n_o}$, where n_o is the total number of objectives, are evaluated to test the quality of the solutions. For the optimization to proceed from the inner loop to the outer loop, consistency must be achieved between the variables at the subsystem level and the system level. The criteria for consistency is stated as $\sum_{i=1}^{N} d_i < 0.01$, where d_i is the consistency between active and copy variables. The strategy of turning the consistency equation into an inequality constraint is also used in other CO-style single- and multi-objective architectures, sometimes using a small variable ε [24, 10, 131].

Before the inner loop is entered again, genetic algorithm operators are used to rank and mutate the solutions to provide diversity and prevent the design points falling into (and staying in) a local minimum.

The termination criteria for the outer loop in PGACO is a pre-defined number of generations. In [8], this is 60.

In the inner loop, the system-level problem is formulated as

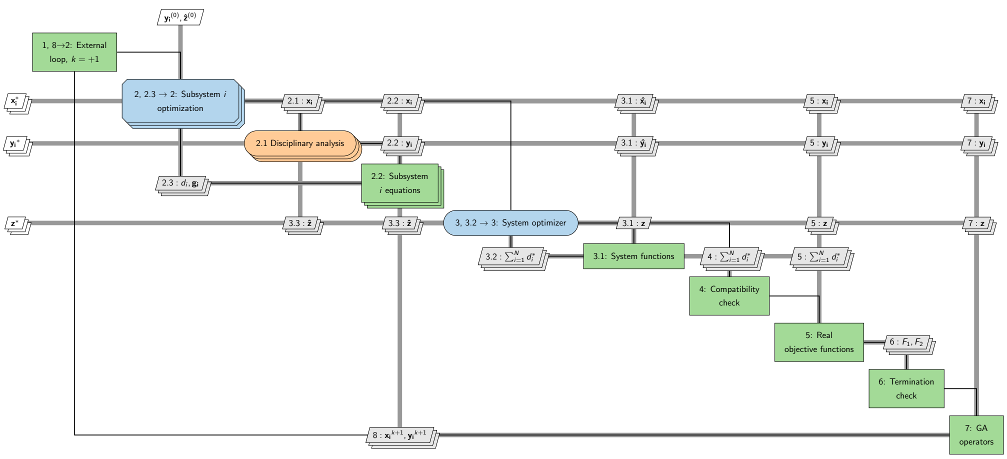
$$
\begin{aligned}
\min \quad & \sum_{i=1}^{N} d_i^* \equiv \sum_{i=1}^{N} \left(\sum_{j=1}^{n_z} (\hat{z}_{i,j} - z_j)^2 + \sum_{j=1}^{n_{y_i}} (y_{i,j} - \hat{y}_{i,j})^2 \right) \\
\text{s.t.} \quad & \hat{\mathbf{y}}, \mathbf{z} \\
\text{w.r.t} \quad & \mathbf{z}^{LB} \leq \mathbf{z} \leq \mathbf{z}^{UB} \\
& \hat{\mathbf{y}}^{LB} \leq \hat{\mathbf{y}} \leq \hat{\mathbf{y}}^{UB}
\end{aligned}
\tag{3.24}
$$

and the subsystem-level i problem is

$$
\begin{aligned}
\min \quad & d_i \equiv \sum_{j=1}^{n_z} (\hat{z}_{i,j} - z_j)^2 + \sum_{j=1}^{n_{y_i}} (y_{i,j} - \hat{y}_{i,j})^2 \\
\text{s.t.} \quad & c_i(\mathbf{x_i}, \mathbf{y_i}, \hat{\mathbf{z}}_i) \leq 0 \\
\text{w.r.t} \quad & \mathbf{x_i}^{LB} \leq \mathbf{x_i} \leq \mathbf{x_i}^{UB} \\
& \mathbf{y_i}^{LB} \leq \mathbf{y_i} \leq \mathbf{y_i}^{UB} \\
& \hat{\mathbf{z}}^{LB} \leq \hat{\mathbf{z}} \leq \hat{\mathbf{z}}^{UB}
\end{aligned}
\tag{3.25}
$$

The XDSM of this method is shown in Figure 3.12. When tested on the multi-objective Sellar problem (see Section 2.2), it is found that PGACO can produce results that are as accurate as those produced by an all-at-once formulation [8].

Figure 3.12: The XDSM diagram of the PGACO method adapted from [8].



Collaborative optimization strategy for multi-objective systems (COSMOS) COSMOS was proposed by Rabeau et al. in 2007 [9]. It is a combination of the collaborative optimization architecture and the use of genetic algorithm components to propose better solutions to the overall problem.

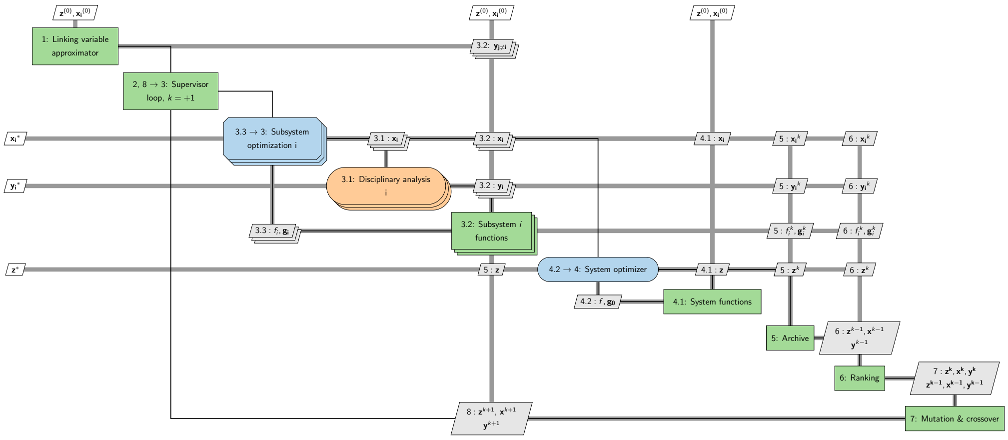
Optimization starts with the subsystems, which takes on a population of local x_i and global z variables. The system-level module approximates the values of the linking variables, and then the subsystem-level optimization is completed using the local variables (treated as design variables). No consistency functions are needed in the COSMOS architecture, because the subsystems do not modify the global variables – these are effectively treated as constants for the purposes of subsystem optimization.

The solutions are sent to the system-level problem, where the memorised and current populations (iterations k and $k-1$ respectively) are concatenated and the best performers are selected based on which solutions Pareto dominate the greatest number of other solutions. The new population is sent back to the subsystem optimizers. Like in [8], the termination criterion for both the system and subsystem optimizers are based solely on whether the allotted number of generations has been fulfilled, although it is acknowledged in the paper that there are more effective approaches to termination criteria than this. The XDSM for COSMOS is given in Figure 3.13, where the current and memorised populations are indicated as x^k and x^{k-1} respectively.

The key difference between COSMOS and the classic CO architecture is that there is are no consistency equations in either the subsystem or the system optimization problems. This is because the local variables in the system and the global variables in the subsystem are treated as constants, not for the optimizers to change; any solutions are automatically constrained by this fact.

A comparison between COSMOS, E-MMGA and MORDACE was undertaken in [68].

Figure 3.13: The XDSM diagram of the COSMOS method adapted from [9].



Architectures using genetic algorithms

Some other a posteriori multi-objective CO architectures do use genetic algorithms to solve the system or subsystem optimization problems, but do not otherwise use features of genetic algorithms to select, rank, mutate (or otherwise) solutions for the next external loop iteration.

E-MMGA Gunawan et al. further developed M-MGA by introducing the entropy-based multi-objective multidisciplinary algorithm (E-MMGA) architecture in [102, 104]. In this case, the entropy index is measured using an influence and density function developed in [132], and is implemented with the intention of ensuring that solutions are evenly spread along the objective space.

This method is useful for the following reasons:

- It is applicable to type one problems, because genetic algorithms are implemented at the subsystem levels
- The diversity of the solutions are maximised via the entropy index, providing good spread of solutions
- Diversity and convergence are monitored during the optimisation process using two quality indices: size of dominated space (see [133]) and entropy.

In E-MMGA, the system-level optimization problem is removed and the subsystem-level problems compensate by including the ex-system constraints,

$$
\begin{aligned}
\min \quad & F_{j,i}(\mathbf{z}, \mathbf{x_i}) \\
\min \quad & \mathbf{f_i}(\mathbf{z}, \mathbf{x_i}) \\
\text{w.r.t} \quad & \mathbf{z}, \mathbf{x_i} \\
\text{s.t.} \quad & \mathbf{g_0}(\mathbf{z}) \leq 0 \\
& \mathbf{g_i}(\mathbf{z}, \mathbf{x_i}) \leq 0
\end{aligned}
\tag{3.26}
$$

where $F_{j,i}$ is an additively separable objective function relating to all subsystems. A useful example given in [102] is that weight is an additively separable function, as the total weight of an engineered product is the sum of all its components.

The E-MMGA process maintains an archive called the *grand pool*, which is updated when the subsystem optimisation processes have been completed. All populations from the subsystems are sent there and concatenated into one large pool. There is also a *grand population*, which is the initial population used for the subsystem optimisation processes. The grand population is made up of the solutions in the grand pool that maximise the entropy index. The authors of [102] suggest utilising a separate optimisation routine, but note that the size of the grand pool can make this prohibitively difficult. Therefore, they also suggest randomised sampling to find the solutions that maximise the entropy.

When compared with other optimisers (VEGA [134], NSGA [135], MOGA [136], MOGA-NA [137]) on the multi-objective speed reducer problem [96], E-MMGA showed better solution diversity than all others, and a larger dominated space than all except NSGA [102].

Approximation-assisted McRO (AA-McRO) AA-McRO is proposed in [138, 139, 83]. There are several key differences between AA-McRO and the original McRO formulation in [140, 141, 12]. Firstly, the 'new McRO' proposed in [83] shifts the system problem down to the lower level and replaces the upper level problem with a co-ordination problem (of the form of a consistency problem, which is how it is referred to from here). The robustness optimization also happens at the lower level.

The consistency problem at the upper level receives the objective and constraint values from the lower level optimizations, and returns the copy and shared variables to the lower level problems. This is done in order to aid convergence.

Secondly, an online approximation routine is added to reduce the computational load. This comes in the form of an online DOE and meta-modelling at the upper level. The metamodels prevent the need for complex models to run many times over, hence making this architecture useful for problems where the associated analyses are particularly complex.

As in the original McRO, robustness is expressed using the vectors of uncontrollable parameters $\mathbf{p}$.

Genetic algorithms approach for multidisciplinary multi-objective collaborative optimization (MCO) Firstly, note that 'MCO' as an abbreviation is not used in the original literature – it is referred to as MOCO, but the methods proposed by Tappeta and Renaud [120] are already called that. Other literature refers to the method in this section as MCO [83], so for clarity, this is referred to as such.

The approach proposed in [10] uses genetic algorithms for the objective functions at both the system and the subsystem level (making this approach appropriate for type six systems). As in [120], auxiliary variables are used to aid the consistency of the linking variables and, in one of the formulations, the global variables. These fulfil the same purpose as the copy variables shown in [24], so the same notation is used here.

Two formulations are proposed in MCO to handle the consistency between variables. In the first formulation, the global variables are treated as parameters in the subsystem problems, and the linking variables are not passed to the system-level problem. The system-level problem for the first formulation is

$$\begin{aligned} \min \quad & \mathbf{f_0} = \mathbf{F}(\mathbf{f_1}, \mathbf{f_2}) \\ \text{w.r.t} \quad & \mathbf{z}, \hat{\mathbf{y}}_{1,\ldots,N} \\ \text{s.t.} \quad & \mathbf{g_0} \leq 0 \end{aligned} \tag{3.27}$$

and the subsystem i problem is formulated as

$$\begin{aligned}
\min \quad & \mathbf{f_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \hat{\mathbf{y}}_j), \quad j = 1, ..., N, \ j \neq i \\
\text{w.r.t} \quad & \mathbf{x}_i, \hat{\mathbf{y}}_i^j \\
\text{s.t.} \quad & \mathbf{g_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \hat{\mathbf{y}}_j) \leq 0 \\
& ||\mathbf{y}_i(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \hat{\mathbf{y}}_j) - \hat{\mathbf{y}}_i||_2 = 0 \\
& ||\hat{\mathbf{y}}_i^j - \hat{\mathbf{y}}_j||_2 = 0
\end{aligned} \tag{3.28}$$

where $\hat{\mathbf{y}}_i^j$ is the copy of the jth linking variable in subsystem i.

In the second formulation, the copies of global variables are used as active design variables at the subsystem problem, with additional constraints ensuring consistency between the global variables/copies at the subsystem and system levels. The system-level problem is formulated as

$$\begin{aligned}
\min \quad & \mathbf{f_0} = \mathbf{F}(\mathbf{f_1}, \mathbf{f_2}) \\
\text{w.r.t} \quad & \mathbf{z}, \hat{\mathbf{y}}_{1,...,N} \\
\text{s.t.} \quad & \mathbf{g_0} \leq 0 \\
& ||\mathbf{y}_i - \hat{\mathbf{y}}_i||_2 = 0, \quad i = 1, ..., N
\end{aligned} \tag{3.29}$$

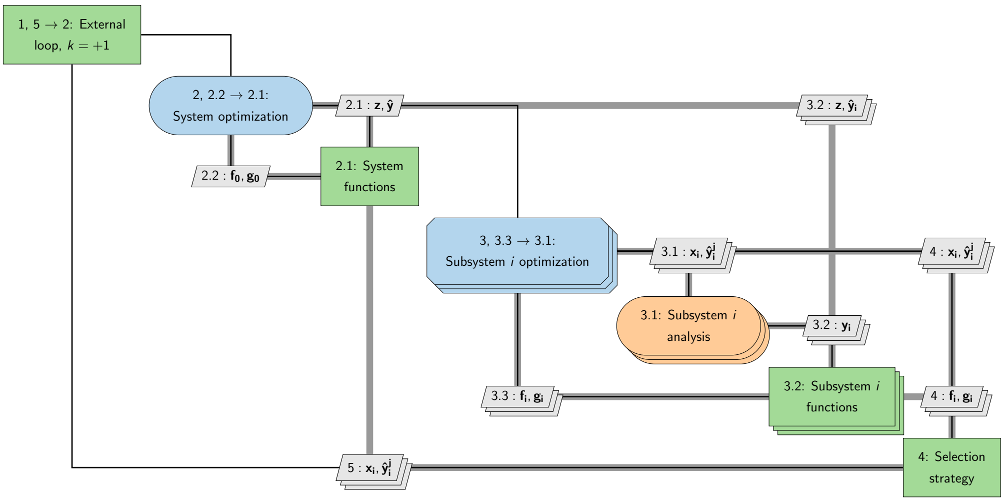
and the subsystem i problem is formulated as

$$\begin{aligned}
\min \quad & \mathbf{f_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \hat{\mathbf{y}}_j), \quad j = 1, ..., N, \ j \neq i \\
\text{w.r.t} \quad & \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \hat{\mathbf{z}}_i \\
\text{s.t.} \quad & \mathbf{g_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \hat{\mathbf{y}}_j) \leq 0 \\
& ||\hat{\mathbf{z}}_i - \mathbf{z}||_2 = 0 \\
& ||\mathbf{y}_i(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \hat{\mathbf{y}}_j) - \hat{\mathbf{y}}_i||_2 = 0 \\
& ||\hat{\mathbf{y}}_i^j - \hat{\mathbf{y}}_j||_2 = 0
\end{aligned} \tag{3.30}$$

In both formulations, it is identified that the one-to-many mapping from the system solutions at iteration k and the solutions at $k+1$ would pose a problem – such an issue is also identified in some of the other CO-style MO-MDO architectures [9, 8, 11]. Four solutions (called selection strategies) to this problem are proposed:

1. Choose the best subsystem solution in a set corresponding to one system-level solution;
2. Choose the worst subsystem solution;
3. Choose a subsystem solution arbitrarily;
4. Combine the solution sets from all disciplines and choose the solutions to move forward based on some fitness metric.

Figure 3.14: The XDSM diagram of the MCO method adapted from [10].



Multi-Objective multidisciplinary genetic algorithms (M-MGA) M-MGA [11] is much closer to the indicator-based approaches found in MOP research than the Pareto approaches usually found in MO-MDO. The system-level problem aims to optimize four quality metrics: maximise hypervolume, spread, and accuracy, and minimise clustering. The system-level problem for M-MGA is

$$\begin{aligned} \min \quad & \mathbf{q}(\mathbf{z}, \mathbf{x}_1^*, ..., \mathbf{x}_N^*) \\ \text{w.r.t} \quad & \mathbf{z} \\ \text{s.t.} \quad & \mathbf{g_0}(\mathbf{z}) \leq \mathbf{0} \end{aligned} \tag{3.31}$$

where $\mathbf{q}$ are the quality metrics described above. The subsystem problems aim to optimize the real-valued functions supplied by the designer. These are expressed as

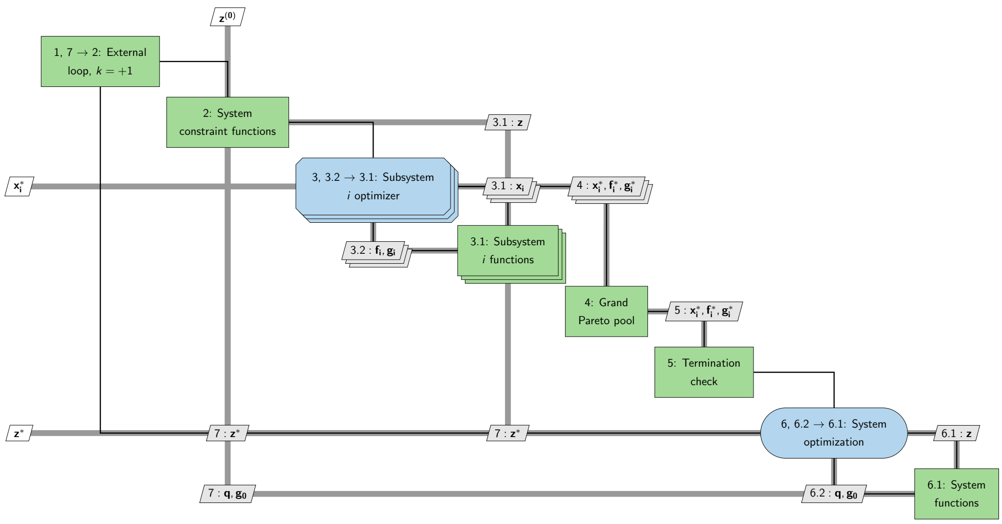
$$\begin{aligned} \min \quad & \mathbf{f_i}(\mathbf{z}, \mathbf{x_i}) \\ \text{w.r.t} \quad & \mathbf{x_i} \\ \text{s.t.} \quad & \mathbf{g_i}(\mathbf{x_i}) \leq \mathbf{0} \end{aligned} \tag{3.32}$$

Like many other multi-objective MDO architectures [9, 8], a key feature of this approach is that it constructs a large set of solutions at the system level. The method used to 'prune' this tends to vary, but in M-MDA, the inferior solutions (i.e. those with a high ranking) are removed to make way for solutions that are considered more 'optimal'.

It can be seen that, while this approach shares similarities with CO in that it is a distributed IDF architecture, consistency is achieved by treating the system variables at the subsystem problem (and vice-versa) as a constant – this concept is also used in the COSMOS architecture [9].

The XDSM of this architecture can be seen in Figure 3.15.

Figure 3.15: The XDSM diagram of the M-MGA method adapted from [11].



Multi-objective robust collaborative optimization (McRO) McRO [12] is for type three MO-MDO problems, with a single-objective co-ordination problem at the system level and multi-objective problems at the subsystem level. It is designed to handle uncertainty using performance sensitivity analyses, developed in [140, 141], with 'uncontrollable parameters' $\mathbf{p_i}$ and $\mathbf{p_z}$ attached to the subsystem and the system problems respectively. Li and Azarm introduce three concepts to aid in the development of McRO:

1. Tolerance region – the region in which the uncertainty is known to vary, expressed verbatim as 'the known interval of uncertainty'.

2. Acceptable objective variance region (AOVR) – the region in which robust designs lie. In other words, when a solution is mapped from the tolerance region to the objective space, a robust design will not breach the AOVR at any point.

3. Acceptable constraint variance region (ACVR) – the region in which feasible robust designs lie. Similarly to the AOVR, when solutions are mapped from the tolerance region to the constraint space, feasible robust designs will not breach the ACVR at any point.

A design can be determined to be robust or feasibly robust using the above measures. This uncertainty is propagated through the subsystems via tolerance ranges in the linking variables, and the copies of linking variables.

Two formulations are proposed in [12]. In both, the global variables are set at the system level and treated as parameters at the subsystem problems. In formulation one, the linking variables are copied between the subsystem and the system problems, and subsystem constraints ensure consistency between all copies.

The subsystem i formulation is

$$\begin{aligned}
\min \quad & \mathbf{f_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \mathbf{p_i}) \\
\text{w.r.t} \quad & \mathbf{x}_i, \hat{\mathbf{y}}_i^j \\
\text{s.t.} \quad & \mathbf{g_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \mathbf{p_i}) \leq 0 \\
& ||\mathbf{y}_i - \hat{\mathbf{y}}_i||_2 \leq \varepsilon \\
& ||\hat{\mathbf{y}}_i^j - \hat{\mathbf{y}}_j||_2 \leq \varepsilon \\
\text{where} \quad & \mathbf{y_i} = \mathbf{Y_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_i^j, \mathbf{p_i})
\end{aligned} \tag{3.33}$$

where $\hat{\mathbf{y}}_i^j$ is the 'copy' of the linking variable from subsystem j produced by subsystem i.

In formulation two, the linking variables are fixed and treated as parameters in the subsystems, similarly to how the global variables are treated. The subsystem i formulation is

$$\begin{aligned}
\min \quad & \mathbf{f_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_j, \mathbf{p_i}) \\
\text{w.r.t} \quad & \mathbf{x}_i \\
\text{s.t.} \quad & \mathbf{g_i}(\mathbf{z}, \mathbf{x}_i, \hat{\mathbf{y}}_j, \mathbf{p_i}) \leq 0 \\
& ||\hat{\mathbf{y}}_i^j - \hat{\mathbf{y}}_j||_2 \leq \varepsilon
\end{aligned} \tag{3.34}$$

For both of the above, the system-level problem is expressed as

$$\begin{aligned}
\min \quad & \mathbf{f_0}(\mathbf{z}, \hat{\mathbf{y}}, \mathbf{p_z}) \\
\text{w.r.t} \quad & \mathbf{z} \\
\text{s.t.} \quad & \mathbf{g_0}(\mathbf{z}, \hat{\mathbf{y}}, \mathbf{p_z}) \leq 0 \\
& \eta_{f,g,C} - 1 \leq 0 \\
\text{where} \quad & \eta_{f,g,C} = max\{\eta_{f,g}, \eta_C\}
\end{aligned} \tag{3.35}$$

where $\eta_{f,g}$ and η_C are derived from the tolerance range and the feasibility and objective robustness described at the beginning of this subsection.

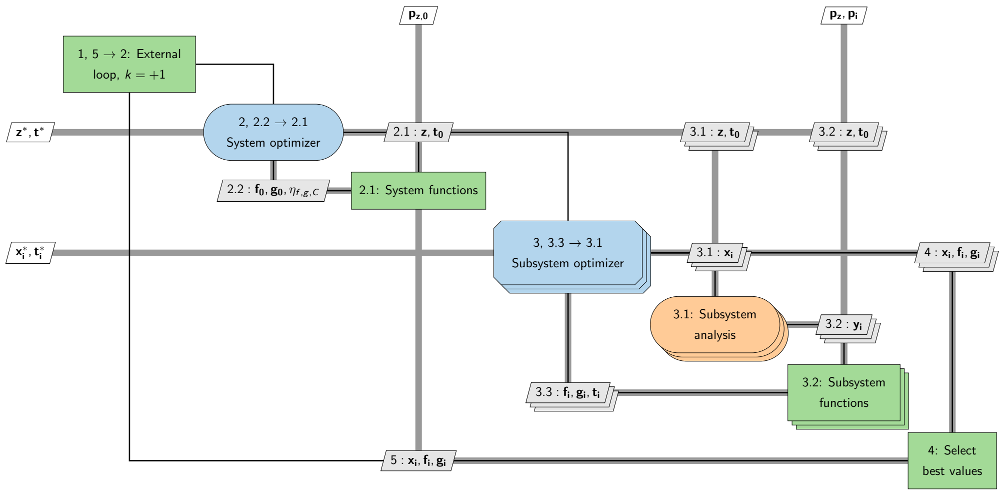
An extension to McRO is proposed in [142] (developed in single-discipline problems in [143]) that has the objective of minimising the investment – or cost –

of finding a solution while also minimising the uncertainty.

Similar, robust MDO methods are covered in [144, 145], however, neither of these methods are CO-style, therefore they will not be covered here.

The XDSM for McRO is shown in Figure 3.16.

Figure 3.16: The XDSM diagram of the McRO method adapted from [12].



An overview of the CO-style MO-MDO architectures

Table 3.2 gives an overview of the key differences and similarities of the different CO-style architectures listed in this sub-section. These include the system type(s) the architecture is able to handle (see Table 3.1), what the system and subsystem objectives are, how the consistency between subsystem and system variables is handled, the selection strategy, and the treatment of the global variables.

Table 3.2: An overview of the key aspects and differences of each of the CO-style MO-MDO methods.

Architecture abbr.	System type	System objective	Subsystem objective	Consistency handling	Selection strategy	Global variable treatment
PGACO	2	Sum of consistency	Consistency	Consistency objectives	Rank	Copies
COSMOS	3	Propose better global variables	Real	Fixed global variables	Best solutions	Fixed
E-MMGA	1	No system objective	Real & additively separable objectives	Fixed global variables	Highest entropy	Fixed
AA-McRO	2	Real	Real & robustness	Consistency objectives (additive to real objective)	No selection	Consistency objectives
MCO	5 or 6	Function of subsystem objectives	Real	Constraints - subsystem and system or fixed global variables	DM's choice	Copies or fixed
M-MGA	6	Metrics	Real	Fixed global variables	Rank	Fixed
McRO	6	Real	Real	Consistency constraints (linking variables)	Best solutions	Fixed

3.4 Discussion and summary

In this section, a discussion is offered on the current state of research in MO-MDO, including key points and identified research gaps. MO-MDO is a small field of research, and is still growing, meaning there are plenty of opportunities to further the research in important aspects. These points are detailed below.

Interactive methods This review has so far covered a priori and a posteriori methods in MO-MDO. The third main approach to facilitate DM preferences is via interactive or progressive methods. This is done via the preferability operator. In these methods, the DM is actively involved in the optimisation process, sometimes by updating weights at intervals throughout the optimisation [146]. These methods may be useful for several reasons, such as reducing the size of the design space that needs to be explored by the optimiser if the DM is interested in a restricted range of solutions [146, 147]. However, it requires the DM to be present throughout the optimisation process. See [148] for a review of interactive methods in multi-objective optimisation.

Despite the importance of interactive and progressive methods in multi-objective optimisation, very little attention has been given to these methods in the MO-MDO research community. Interactive methods allow decision-makers to be involved in the optimisation process, but all methods found in the corresponding literature consign the DM to a role before or after optimisation. Some future work should be dedicated to developing such architectures for use in MO-MDO.

Replication While there are many methods that are designed to handle multi-objective MDO problems outlined in this review, in the related literature there are very few instances of replication or use in applied problems. It would be good to see further literature either comparing the performances of some of these architectures, or simply using them in test problems, particularly problems with features that are difficult to handle such as multi-modality and large numbers of disciplines.

More replication studies using new, large or complex problems in MO-MDO would unearth further strengths and weaknesses of the different formulations, and lead to the improvement of these methodologies.

Treatment of linking variables Linking variables are important in MDO formulations because they allow data to flow between the subsystems in the problem. As a result, it is critical that how they are handled in their formulations are specified properly. Linking variables, their generation, and their treatment, are frequently neglected in the MO-MDO literature [132, 104, 102, 68]. In some cases, formulations and methodologies do not explicitly state how linking variables are handled, such as in [120], while others seem to ignore them completely [57].

Lack of interest in newer MDO architectures The architectures that have been subject to the changes necessary to handle MO-MDO problems are primarily the oldest architectures (i.e. CO and CSSO [24]). While it is understandable that these architectures would garner the most attention, more time and effort needs to be dedicated to architectures such as enhanced collaborative optimisation [149] and asymmetric subspace optimisation [150].

Benchmarking in MO-MDO Despite the progression of MDO and MO-MDO architectures and strategies, the academic benchmarks have struggled to keep up, with the applied problems seemingly the preferred method of testing and architecture showcasing in most works. Not only do most of the academic problems fail to meet the requirements of scalability and complexity that make them comparable to or even useful for real-world applications, other problem features have been ignored entirely. Huband et al. specified seven recommendations for the construction of multi-objective test suites [89]: no extremal parameters, no medial parameters, a scalable number of parameters, a scalable number of objectives, dissimilar parameter domains, dissimilar trade-off ranges, and known optima. Additionally, they recommended five features of multi-objective optimisation problems: a varied Pareto-optimal geometry (including convex, concave, mixed, disconnected, degenerate Pareto fronts), objectives that are either separable or non-separable, bias, many-to-one mappings, and multi-modality [89]. On top of recommending the addition of a scalable number of disciplines, authors of future benchmarks for MO-MDO architectures should consider how the recommendations made in [89] affect MDO problems.

The NASA MDO test suite provides a set of problems that are simple, and in some cases, linearly separable, although it seems not all of these problems have been taken advantage of in the MO-MDO sphere, particularly the heart dipole and propane combustion problems [94]. Reformulating these into multi-objective problems, constructing a full test suite and analysing the resulting problems in the style of Huband et al.'s benchmarking review would be a good choice for future research directions.

While there is no singular definition of what makes a good benchmark problem in optimization research, there are some criteria that should be considered important when evaluating different algorithms for performance. In MO-MDO, these may include:

- Scalability in the number of objectives
- Scalability in the number of variables
- Scalability in the number of disciplines
- Variability in the Pareto front geometry, including convex, non-convex, mixed and discontinuous (for example, as suggested in Huband et al [89])
- Variability in the complexity of the relationships between disciplines and topologies
- Extendability to alternate MDO architectures, such as those described in [24]. This is particularly important in the case of extension to both monolithic and distributed architectures.
- Variability in the difficulty of the MDA problem.

This Chapter has discussed and evaluated some of the key issues affecting the field of MO-MDO research, including key concepts, methods and benchmarks. Key concepts that have been introduced were architectures, concepts used in MDO and MOPs, and methods frequently used in MO-MDO research such as game theory, metamodels and problem partitioning. Also introduced was a typology for categorising MO-MDO methods. A priori and a posteriori MO-MDO methods were introduced and XDSM diagrams were used in conjunction with optimisation formulations to illustrate their processes. Academic and applied benchmark problems were introduced, and some of the most common benchmark problems found in the literature were introduced. Finally, a discussion of the key information derived from the literature review was presented.

One of the main discussion points from this review is addressed in Chapter 4: There are insufficient scalable MO-MDO benchmark problems to serve the field adequately. This is done by proposing, testing and evaluating a scalable MO-MDO benchmark problem using PyOptSparse [151] and OpenMDAO [152], a popular software in the MDO community.

Chapter 4

Toward scalable benchmark problems for multi-objective multidisciplinary optimization

4.1 Introduction

Benchmarking is an important aspect of optimization research. It allows researchers and industry specialists to evaluate and analyse the performance of optimization algorithms, and is particularly useful when the benchmark problems are complex, similar to real-world problems, or otherwise challenging to the optimizer.

One of the key findings reported in the literature review in Chapter 3 was that scalable benchmarking is an understudied area of MO-MDO research and development – in other words, there are no readily-available MO-MDO scalable benchmark problems. This is despite the importance of multi-objective problems in practical applications where MDO might also be useful or even currently in use.

As a result, it is important that a multi-objective benchmark problem for use in MDO is developed. This should also be scalable according to the needs of the optimization algorithm or MDO architecture being tested.

This chapter aims to identify, discuss and investigate some of the issues arising from scaling multi-objective MDO problems in the number of disciplines by modifying the well-known ZDT1 test problem incorporating the approach used by Tedford and Martins [48] for single-objective problems. This benchmark problem is evaluated using a common MDO architecture and the results are analysed and discussed. The separability of the ZDT1 problem is exploited in this benchmark. The MDO-ZDT1 benchmark test problem is scalable in number of disciplines and variables which can be used to evaluate the performance of algorithms that handle bi-objective MDF problems. Additionally, it is separable which is useful for the conversion into an MDO problem.

Scaling the number of disciplines in the problem, implying an increasing number of decision variables, does produce substantive changes in convergence ability. It is also shown that the accuracy of the MDA solver has an impact on the convergence ability of the multi-objective optimization algorithm, which is particularly noticeable when moving from 7 to 14 disciplines. Modification of standard (non-MDO) multi-objective benchmark problems is a promising approach to developing scalable multi-objective MDO benchmarks.

This chapter is organised as follows. An introduction to MDO architectures, with special focus on the MDF approach, is provided in Section 4.2. A proposed multi-objective MDO problem with scalable number of disciplines based on the ZDT1 test problem is described in Section 4.3. The experimental setup used in this chapter is in Section 4.4. The experimental results in Section 4.5 includes the effects of introducing varying number of disciplines to the problem (Section 4.5.2), and the influence of solver accuracy on convergence (Section 4.5.3). The chapter concludes with a summary in Section 4.6.

4.2 The MDF architecture

The multidisciplinary feasible architecture comes under the umbrella of monolithic architectures in MDO. Monolithic architectures are those where only one optimization problem is present, but disciplinary analyses are used as part of that problem. In MDF, these analyses interact directly with each other, using the linking variables from other analyses to obtain a solution. These solutions must be consistent with each other before the optimization can proceed.

The MDF architecture is included here for clarity. The optimization problem is expressed as

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathbf{z}) \\ \text{w.r.t} \quad & \mathbf{x}, \mathbf{z} \\ \text{s.t.} \quad & \mathbf{c_0}(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathbf{z}) \leq 0 \\ & \mathbf{c_i}(\mathbf{x}_i, \mathbf{y}_i(\mathbf{x}_i, \mathbf{y}_{j\neq i}, \mathbf{z}), \mathbf{z}) \leq 0, \qquad i = 1, ..., N \end{aligned} \tag{4.1}$$

For a total of N disciplines $\mathbf{x}$ is a vector that contains all local variables, that is, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$, and at the ith discipline ($i \in \{1, \ldots, N\}$) there are n_{x_i} local variables given by the vector $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n_{x_i}})^T$. A total of n_z global variables are in the vector $\mathbf{z} = (z_1, \ldots, z_{n_z})^T$. Similarly to $\mathbf{x}$, the vector $\mathbf{y}$ contains all linking variables per discipline, that is, $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T$, and at the ith discipline there are a total of n_{y_i} linking variables given by the vector $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,n_{y_i}})^T$. $\mathbf{c}_0$ and $\mathbf{c}_i$ are both vectors representing the global and local constraints, respectively, and f_0 is the objective function.

One of the characteristics of the MDF architecture is that each discipline is solved in turn, often by the use of the Gauss–Seidel multidisciplinary analysis (MDA) procedure [153] or by the use of Newton-based methods. This implies that the different discipline analyses cannot be conducted in parallel but it ensures consistency between the linking variables inputs and outputs (this is particularly important if there are circular dependencies between the disciplines as is the case in this chapter's problem formulation). One of the drawbacks of this architecture is that a full MDA needs to be conducted for each candidate design, and the MDA needs to be run until a consistent set of linking variables are found. Not having a consistent set of linking variables after each analysis can have an impact on the convergence of the optimization algorithm, as demonstrated in this chapter.

4.3 Proposed MDF MDO-ZDT benchmarking problem

For clarity, a separate notation is established for the design variables in the problem. The decision vector is $\mathbf{w} = (w_1, \ldots, w_{n_v})^T$, and the original ZDT1 problem is expressed as:

$$\begin{aligned} \min \quad & f_1(\mathbf{w}), f_2(\mathbf{w}) \\ f_1(\mathbf{w}) &= w_1 \\ f_2(\mathbf{w}) &= g(\mathbf{w})h(f_1(\mathbf{w}), g(\mathbf{w})) \\ g(\mathbf{w}) &= 1 + \frac{9}{n_v - 1}\left(\sum_{i=2}^{n_v} w_i\right) \\ h(f_1(\mathbf{w}), g(\mathbf{w})) &= 1 - \sqrt{\frac{f_1(\mathbf{w})}{g(\mathbf{w})}} \end{aligned} \tag{4.2}$$

where n_v is the number of decision variables, and $0 \leq w_i \leq 1$ for all $i = 1, \ldots, n_v$. The Pareto optimal solution set for the ZDT1 problem is given by

$$f_2(\mathbf{w}) = 1 - \sqrt{f_1(\mathbf{w})}. \tag{4.3}$$

The optimal Pareto front for the non-MDO ZDT1 problem is shown in Figure 4.1. In this solution, the leading variable w_1 varies between 0 and 1, while all other variables are 0.

The ZDT1 problem is used here to explore scalability in disciplines within certain bounds. These bounds are dictated by the user of the benchmark via the chosen number of variables n_v, number of design and linking variables for each discipline, and the total number of disciplines. In this study, each discipline must contain at least one local variable and one linking variable, and there must be at least one global variable. However, some considerations first must be established to illustrate the constraints of the problem. The original ZDT1 problem contains n_v design variables, and all design variables except the first, w_1, are separable in the $g(\mathbf{w})$ function (Equation 4.2). This means that values in this function can be exchanged freely,

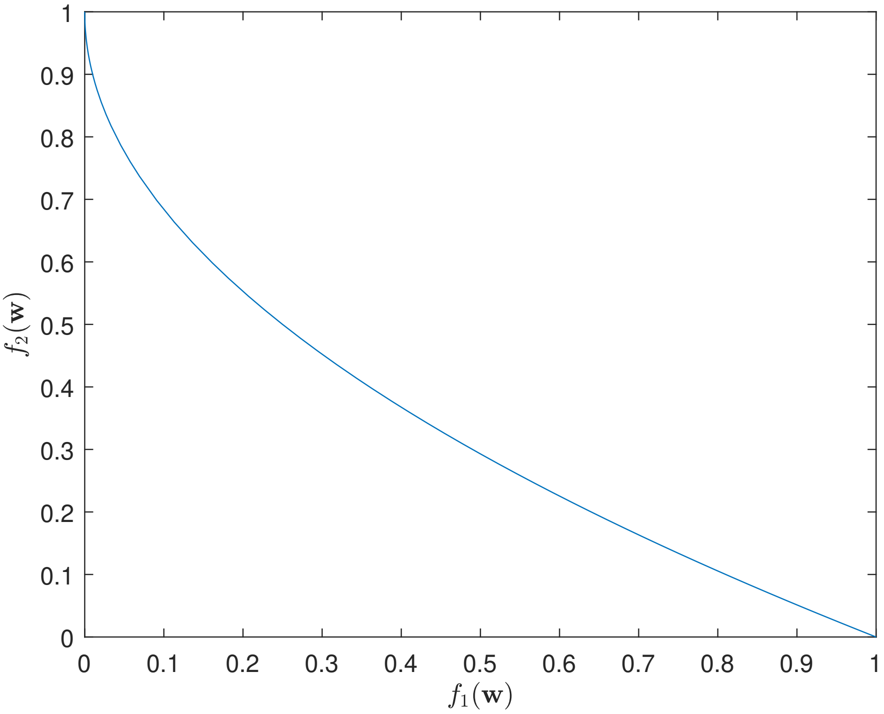


Figure 4.1: The Pareto front of the original, non-MDO ZDT1 problem.

without worrying about the effect of variable sequencing on either the disciplinary analyses or the objective functions, as long as w_1 remains the same. Extending this problem via the n_v variable allows the number of design variables to have no upper bound. As a result, it is simple to exchange both linking, global and local design variables for the design variables in the original ZDT1 problem by simply summing all global (excluding z_1), local and linking variables in the $g(\mathbf{x}, \mathbf{y}, \mathbf{z})$ function. Additionally, there is no possibility for interactions between same-discipline local and linking variables that would prevent the optimizer from converging. The optimization problem is given in (4.4) and the MDA equations are given in (4.5).

$$
\begin{aligned}
\min \quad & f_1(\mathbf{z}), f_2(\mathbf{x},\mathbf{y},\mathbf{z}) \\
& f_1(\mathbf{z}) = z_1 \\
& f_2(\mathbf{x},\mathbf{y},\mathbf{z}) = g(\mathbf{x},\mathbf{y},\mathbf{z})h(f_1(\mathbf{z}), g(\mathbf{x},\mathbf{y},\mathbf{z})) \\
& g(\mathbf{x},\mathbf{y},\mathbf{z}) = 1 + \frac{9}{n_v}\left(\sum_{i=2}^{n_z} z_i + \sum_{i=1}^{N}\sum_{j=1}^{n_{x_i}} x_{i,j} + \sum_{i=1}^{N}\sum_{j=1}^{n_{y_i}} y_{i,j}\right) \\
& h(f_1(\mathbf{z}), g(\mathbf{x},\mathbf{y},\mathbf{z})) = 1 - \sqrt{\frac{f_1(\mathbf{z})}{g(\mathbf{x},\mathbf{y},\mathbf{z})}} \\
& \text{s.t.} \quad 0 \le z_i \le 1, \quad i = 1, \dots, n_z \\
& 0 \le x_{i,j} \le 1, \quad i = 1, \dots, N \text{ and } j = 1, \dots, n_{x_i} \\
& 0 \le y_{i,j} \le 1, \quad i = 1, \dots, N \text{ and } j = 1, \dots, n_{y_i} \\
& \sum_{i=1}^{N} n_{x_i} + \sum_{i=1}^{N} n_{y_i} + n_z = n_v + 1
\end{aligned}
\tag{4.4}
$$

where

$$
\mathbf{y}_i(\mathbf{x}_i, \mathbf{y}_j, \mathbf{z}) = -D_i^{-1}(A_i\mathbf{x}_i + C_i\mathbf{z} - B_i\mathbf{y}_j), \text{ and } j = i - 1. \tag{4.5}
$$

Following the approach in [48], the MDA consists of a system of linear equations, and solving them determines the value of the linking variables for each discipline. The inputs to these equations are the local design variables, global design variables, and the linking variables obtained from the other disciplines. Weights equations A, B, C and D are used to customise the equations for each discipline and have dimensions $n_{x_i} \times n_{y_i}$, $n_{y_j} \times n_{y_i}$, $n_z \times n_{y_i}$ and $n_{y_i} \times n_{y_i}$ respectively. Matrix D must be invertible, and in this chapter is an identity matrix.

The un-processed matrices are notated with a bar, $\bar{\ }$. Matrices $\bar{A}$, $\bar{B}$ and $\bar{C}$ are randomly generated, with values varying between 1 and 11. This is to prevent any matrix elements from having a value of zero, while providing a range of 10 for non-normalised values. The rows of these weights matrices are then normalised to sum to 1 by using Equation 4.6 for matrix A, Equation 4.8 for matrix B and

Equation 4.7 for matrix C. j and k are the element numbers of each matrix.

$$A_{i,j,k} = \frac{\bar{A}_{i,j,k}}{\sum_{j=1}^{n_{x_i}}(\bar{A}_{i,j,k})} \tag{4.6}$$

$$C_{i,j,k} = \frac{\bar{C}_{i,j,k}}{\sum_{j=1}^{n_z}(\bar{C}_{i,j,k})} \tag{4.7}$$

The rows of the B matrix are multiplied by 2 to enforce the bound constraints on the linking variables within the MDA equations.

$$B_{i,j,k} = \frac{2\bar{B}_{i,j,k}}{\sum_{j=1}^{n_{y_i}}(\bar{B}_{i,j,k})} \tag{4.8}$$

While it is possible to include the linking variables from more than one discipline, for simplicity, a circular relationship between the preceding disciplines has been adopted.

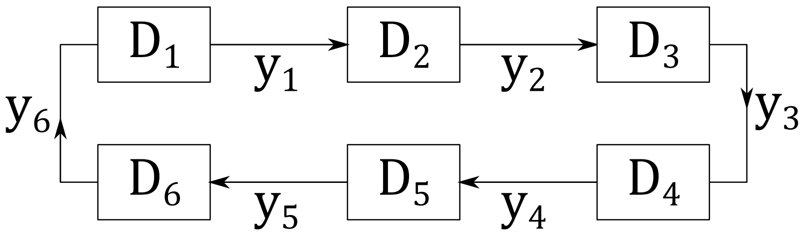


Figure 4.2: An example of the linking relationships in a 6-discipline system, where D_i represents the subsystem number.

Figure 4.2 is given as diagrammatic example of the disciplinary relationships used in Equation 4.5, for a six-discipline problem. This was chosen because it is easily scalable for all design and linking variables, as well as number of disciplines, and guarantees convergence.

4.4 Experimental setup

The experiments were undertaken using the OpenMDAO [152] and PyOptSparse [151] packages in Python. The MDF architecture was used to frame the problem. The non-dominated sorting genetic algorithm 2 (NSGA-II) algorithm [111] has been used to solve the optimization problem in (4.4). The default settings suggested by the pyOptSparse software[1] are used, a population size of 100, crossover and mutation probabilities set to 60% and 20%, respectively. Based on initial experiments good convergence has been observed after 100 generations, and this has been set as the termination criterion. Two solvers were used in the MDA routines—a Newton nonlinear solver with an iteration limit of 1000 and no relaxation factor, and a direct solver. Both were provided by the OpenMDAO Python package. For the purposes of statistical significance, each experiment was run 21 times, and the run with the median inverted generational distance (IGD) was chosen for analysis.

In this chapter, the total number of design and linking variables are modified to maintain similarity in subsystem size across all disciplines. In this case study, each discipline contains 5 linking variables. These values are specified in Table 4.1. For the sake of simplicity, the number of linking and local design variables in each discipline is the same, i.e. $n_{x_i} = n_{y_i}$. The initial values for the design variables were randomly generated values between 0 and 1. These values are used to initialise the solver-optimizer cycle in OpenMDAO. The processed A, B and C matrices as used in these investigations are available in the project's GitHub repository[2].

Table 4.1: Experimental setup for the modified ZDT1 problem.

Number of disciplines	**Number of global variables**	**Number of local & linking variables per sub-problem**	**Total variables per problem**
$\mathbf{N}$	$\mathbf{n_z}$	$\mathbf{n_{x_i}, n_{y_i}}$	$\mathbf{n_v} + 1$
2	10	5, 5	30
7	10	5, 5	80
14	10	5, 5	150

To evaluate the quality of solutions sets, the following indicators are used:

1. Hypervolume metric: used to measure both the convergence towards and diversity across the Pareto front. The metric quantifies the volume defined by the boundary between the obtained solution set at each generation, and a reference point. The reference point is set to $(1.0, 7.0)$ for all runs. To determine

[1]pyOptSparse provides an implementation of NSGA-II and the default settings are shown in `https://mdolab-pyoptsparse.readthedocs-hosted.com/en/latest/optimizers/NSGA2.html`.

[2]github.com/vj2Sheffield/SSCI_MDO_test_problems

the hypervolume a dimension-sweep algorithm is used [154].

2. Convergence of the variables (global, linking and decision variables): the Tchebychev scalarisation metric, defined by $\text{argmin}_i \ \max_j \ \alpha_j f^i_j$, is used to select the solution from amongst the available alternatives across the Pareto front. α is set to 0.5 in order to target a point in the middle of the Pareto-optimal front.

3. Differences between the empirical attainment functions (EAF) [155] when comparing two algorithms across multiple runs. This metric provides visual information of where an algorithm has done better than the other across the Pareto front.

4.5 Experimental results

4.5.1 Varying the number of variables in non-MDO ZDT1

Firstly, the non-MDO ZDT1 problem was solved using the NSGA2 engine and the settings specified in Section 4.4. This allows a fair comparison between the MDO and non-MDO problems in convergence, solutions, and other performance indicators.

Figure 4.3 shows the convergence of the design variables (excluding the first design variable) for the non-MDO ZDT1 problem for 30, 80 and 150 design variables. This is illustrated here as a benchmark of how well the variables converge after 100 generations, to be compared with the MDO ZDT1 problem shown in Sections 4.5.2 to 4.5.3. These quantities of variables were chosen to match up with the total number of variables per problem shown in Table 4.1.

It can be seen that the design variables are closer to zero by the 100th generation in the 30-variable problem, Figure 4.3a than in the 80- and 150-generation problems. In the 80-variables problem, Figure 4.3b, the design variables tend to settle between 0 and 0.05 by the 100th generation. In the 150-variables problem, Figure 4.3c, the design variables tend to settle between 0 and 0.1 by the 100th generation, demonstrating a clear degradation of convergence qualities as the number of design variables increases.

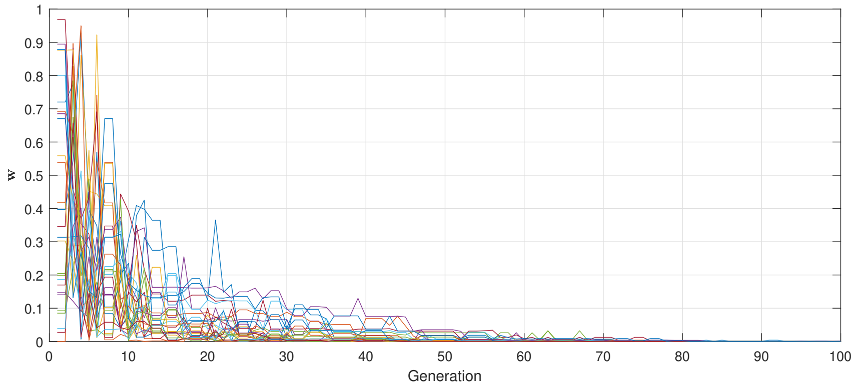
Figure 4.4 shows the solutions obtained by the optimizer for the non-MDO ZDT1 problem at the 100th generation with the varying number of design variables. The 30-, 80- and 150-variable problems are shown in blue, red and yellow respectively. It can be seen that the 30-variable problem produces a set of solutions that are closest to the Pareto front, followed by the 80-variable problem, and then the 150-variable problem. This matches up with the results shown in Figure 4.3, as by the 100th generation, not all design variables in the 80- and 150-variable problem had fallen close to zero.

4.5.2 An increase in the number of disciplines has a deteriorating impact on the convergence of the optimization algorithm

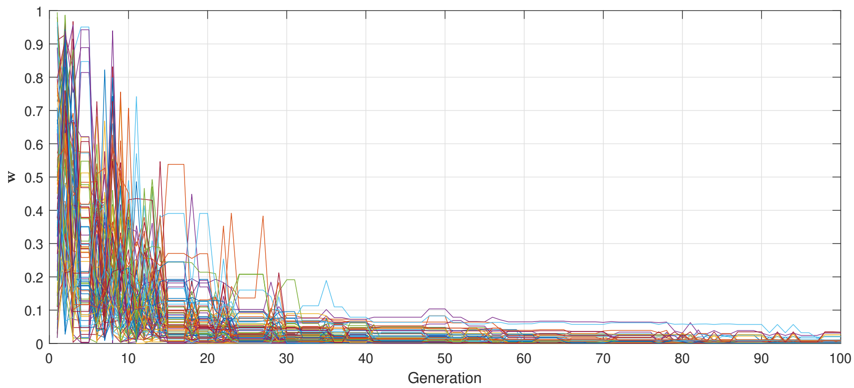
In this section, the effect of scaling the number of disciplines, and therefore also the number of variables, in the convergence of the optimization algorithm when applied to the problem (Equation 4.4) is studied. The obtained results in Figure 4.5 show that the convergence deteriorates with an increase in the number of disciplines from 2 to 14, in that:

1. The differences between the EAFs is shown in Figures 4.5a, 4.5b, 4.5c, and the magnitude of the EAFs differences clearly favours the problems with fewer disciplines.
2. The hypervolume metric captured along the generations in Figure 4.6 shows that a lower number of disciplines means a faster convergence, and also that the variance across runs remains stable with an increase in the number of disciplines.

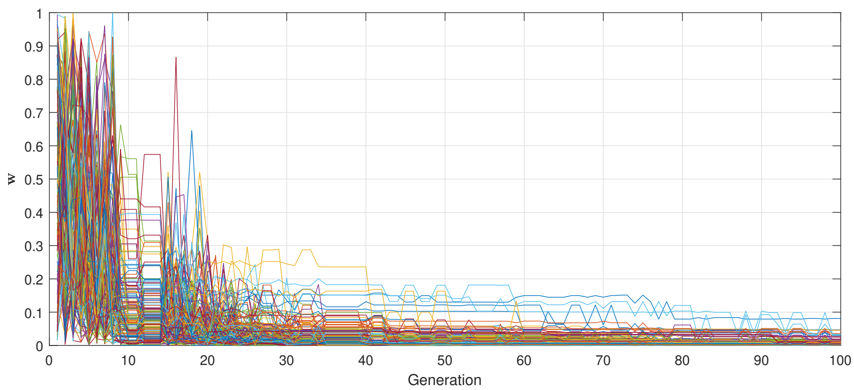
Figure 4.7 shows the convergence of the global, local and linking variables along the optimization run. This is shown separately for the two-discipline, seven-discipline and 14-discipline problem in Figures 4.7a, 4.7b and 4.7c. In the two-



(a) 30 design variables in non-MDO ZDT1

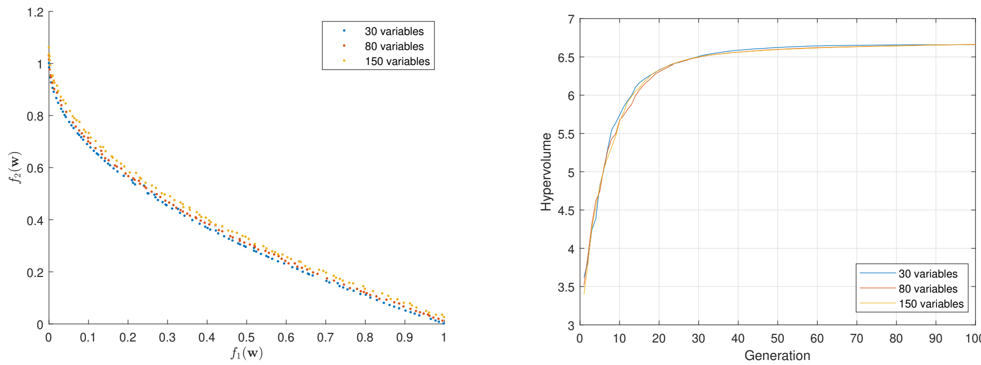


(b) 80 design variables in non-MDO ZDT1



(c) 150 design variables in non-MDO ZDT1

Figure 4.3: The convergence of design variables in the non-MDO ZDT1 problem for 30, 80 and 150 design variables, excluding w_1.



(a) The solutions in the objective space obtained by the optimizer at the 100th generation. The 30-, 80- and 150-variable problems are shown in blue, red and yellow respectively.

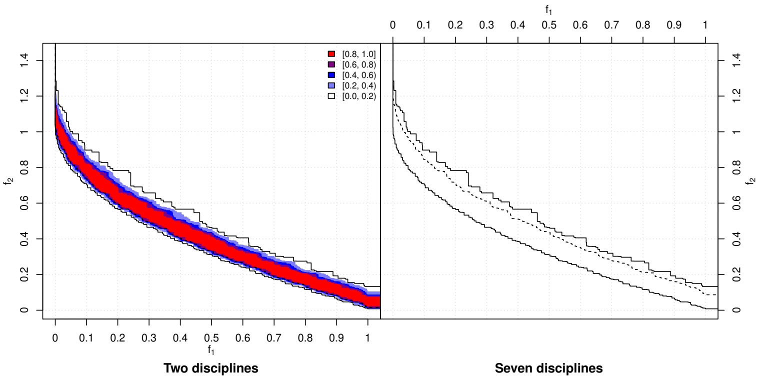
(b) Hypervolumes

Figure 4.4: The solution sets and hypervolumes obtained by the NSGA2 algorithms for the non-MDO ZDT1 problem.

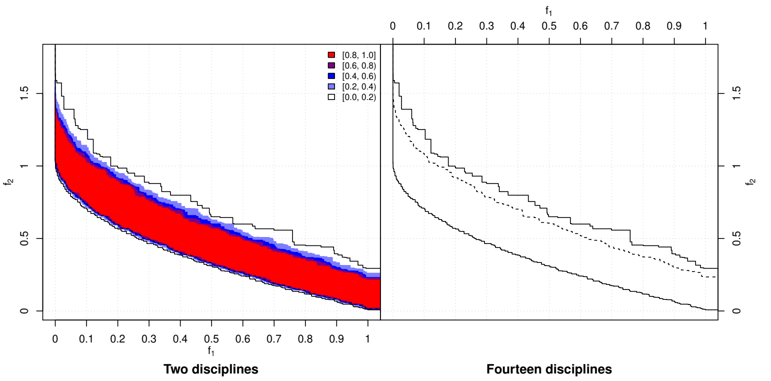
discipline problem, it can be seen that all variables except z_1 converge towards 0 by the 100th generation, while the global variable z_1 converges towards 0.08. In the seven-discipline problem, the linking variables converge towards 0, but have a much shallower curve than in the two-discipline problem. The local variables also deviate further from 0 than they did in the two-discipline problem. z_1 converges towards 0.08, and the global variables approach 0 as they did in the two-discipline problem. In the fourteen-discipline problem, many of the variables have failed to settle by the 100th generation, with the linking variable values varying between 0 and 0.1 by the end of the run. The local variables values vary between 0 and 0.6 by the end of the run, and the global variables between 0 and 0.3. z_1 converges towards 0.3. It is clear that as the number of disciplines increases, it takes longer for the optimization algorithm to converge towards the optima.

4.5.3 The MDA solver accuracy has a higher impact on the convergence with an increase in the number of disciplines

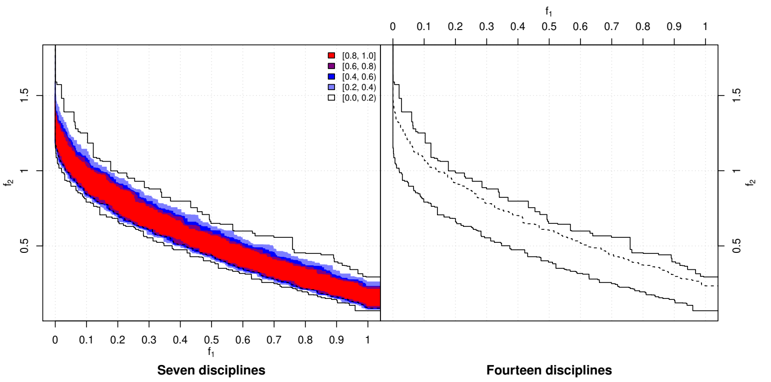
In this section, the impact that the accuracy of the MDA solver can have



(a) EAF: Two versus seven disciplines



(b) EAF: Two versus fourteen disciplines



(c) EAF: Seven versus fourteen disciplines

Figure 4.5: Influence of the number of disciplines on the convergence of the optimizer. The non-dominated solutions obtained at the end of the optimization run, for a total of 21 runs, are compared by using the empirical attainment function. For each subfigure, the plot in the left highlights the differences in favour of case 1, and the plot in the right highlights the differences in favour of case 2. The colour level encodes the magnitude of the observed differences. The lines in the left, centre and right correspond to the best, median and worst attainment surfaces, respectively.

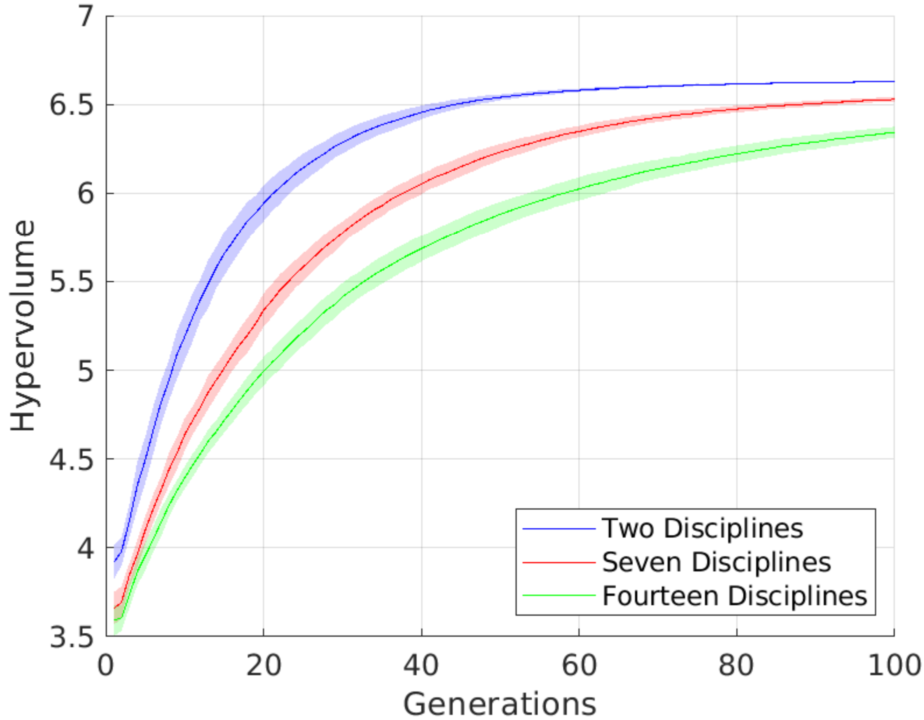
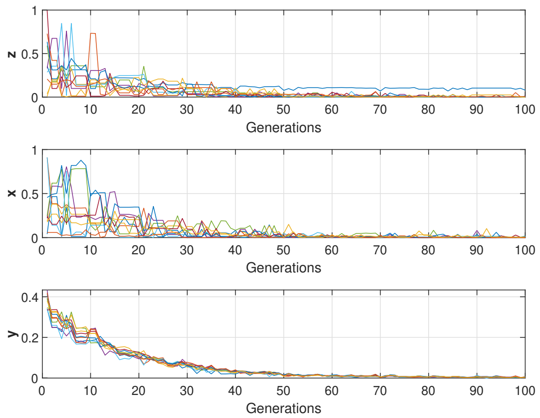


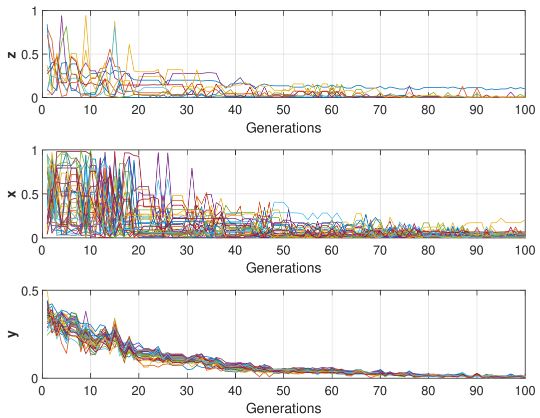
Figure 4.6: The approximation to the true Pareto-optimal front is measured by the hypervolume metric (the higher the better). The shaded region is the standard deviation across runs

on the convergence of the optimization algorithm is studied. In the previous section the maximum number of iterations for the Newton method, used to solve the MDA, was set to 1000. In here it is set to only 2. The obtained results are shown in Figure 4.8 and indicate that the convergence deteriorates with an increase in the number of disciplines, as captured by the differences of the empirical attainment functions (EAFs). The deterioration in terms of performance is particularly visible for the 14-discipline case as shown in Figure 4.8c, but can be negligible for cases with a lower number of disciplines, as is the case with 2 and 7 disciplines (Figures 4.8a and 4.8b, respectively).

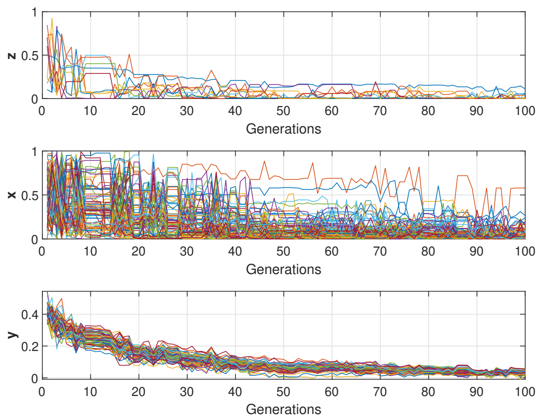
This result is expected, given that as more disciplines are involved in the MDA, the system of equations that needs to be solved is larger. As the number of equations increases, the MDA solver is likely to require a higher number of iterations to ensure good convergence. It has been shown that if the MDA solver lacks convergence, it can have an impact on the ability of the multi-objective optimization



(a) Convergence: Two disciplines

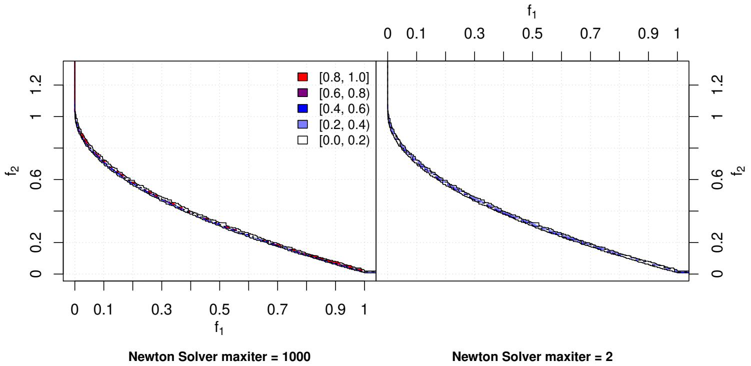


(b) Convergence: Seven disciplines

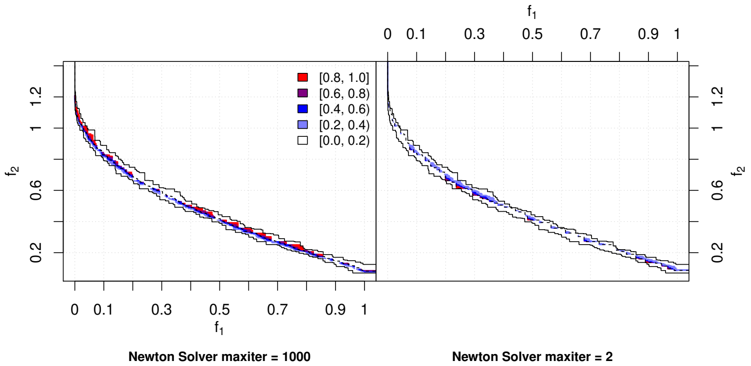


(c) Convergence: Fourteen disciplines

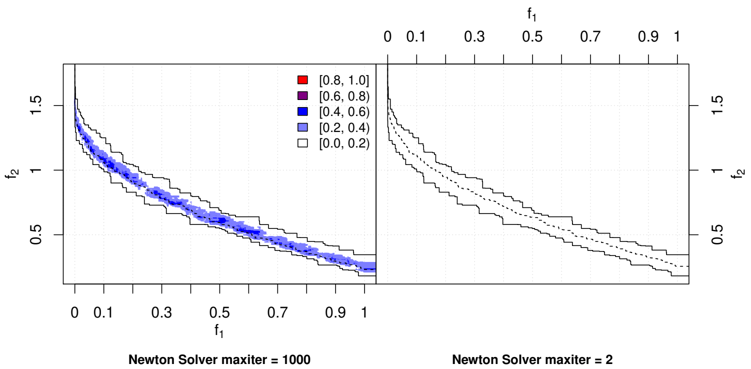
Figure 4.7: Monitoring the convergence of the decision variables along generations. In each subplot, convergence is shown for (a) global design variables, (b) local design variables, and (c) linking variables.



(a) Two disciplines



(b) Seven disciplines



(c) Fourteen disciplines

Figure 4.8: Influence of the MDA solver accuracy on the optimizer's convergence. The maximum number of iterations used by the Newton's method to solve the MDA has been set to 1000 and 2, corresponding respectively to high accuracy and low accuracy. The non-dominated solutions obtained at the end of the optimization run, for a total of 21 runs, are compared by using the empirical attainment function. For each subfigure, the plot in the left highlights the differences in favour of case 1, and the plot in the right highlights the differences in favour of case 2. The colour level encodes the magnitude of the observed differences. The lines in the left, centre and right correspond to the best, median and worst attainment surfaces, respectively.

algorithm to find a good approximation to the Pareto-optimal front.

4.6 Summary

The practical problems faced in the engineering domain often have more than one objective, and in questions of design and decision-making, there is almost always some trade-off. This fact has been largely overlooked by researchers in MDO, especially benchmarking specialists, who have thus far focused mainly on single-objective MDO problems. Furthermore, the existing test problems lack scalability, which is an important aspect of benchmarking in optimization. In this chapter, a benchmark problem that is scalable in variables and disciplines has been introduced, which is based on the non-MDO ZDT1 problem.

Two key aspects were considered in this chapter. The first one was the effect of increasing the number of design variables and disciplines. For the purposes of comparison, the non-MDO ZDT1 problem was also optimized with the same number of variables and optimization engine as those in the MDO ZDT1 problem. While the non-MDO optimization results do show a greater departure from the optimal design variables as the number of design variables increases, this effect is smaller than that seen in the MDO-ZDT1 problem. This indicates that it is the interaction between greater numbers of disciplines that is responsible for the larger deviations from the Pareto optimal front – this is an important consideration for practitioners and researchers in the field of MO-MDO.

The second key aspect was the effect of modifying the number of iterations the MDA solver was allowed to complete before termination. It was found that limiting the number of solver iterations negatively impacted the non-dominated solutions found by the optimizer. While it is always a good idea to allow solvers an adequate number of iterations to reach an accurate solution, it is not always possible to ensure a large number of solver iterations (for example in very large problems where computational resources or time are limited). It is important that practitioners and researchers are aware of the effect that a limited number of solver iterations can have on the solutions generated in an MO-MDO problem.

In Section 3.4, a set of suggested criteria for a good benchmark in MO-MDO is detailed. The ways in which the performance of the ZDT1-MDO problem with respect to these criteria is given below:

- Scalability in the number of objectives: This is not satisfied, as the problem is based on the bi-objective ZDT1 problem (i.e. it has two objectives across all problem instances).
- Scalability in the number of variables: This is met, in both local, linking and global variables.
- Scalability in the number of disciplines: The number of disciplines is scalable in the ZDT1-MDO problem.
- Variability in the Pareto front geometry: The ZDT1-MDO problem only has one Pareto front, which is convex and continuous.
- Variability in the complexity of the relationships between disciplines and topologies: Only one relationship between disciplines is given in this problem formulation. That is, each discipline only takes the linking variables from one other subsystem, and outputs to one other subsystem.
- Extendability to alternate MDO architectures: The problem developed in this chapter is extendable to other architectures, but this may not always be trivial to achieve.
- Variability in the difficulty of the MDA problem: Only one MDA problem has been used in this chapter, and it is a linear problem. This is a useful consideration for future work.

In summary, both increasing the number of disciplines and the number of variables (design and linking), and reducing the number of permitted solver iterations can decrease the quality of solutions in the MDO-ZDT1 optimization problem, or in other words, the closeness of the non-dominated solutions to the Pareto front. Scaling up the problem has a negative effect on the solutions when the NSGA-II parameters are kept the same, with the same number of generations and size of population.

In addition to scalability, there are other features of optimization problems that should be included in a good benchmark test suite for MO-MDO problems. One of these is the complexity of the relationships between the disciplines. In this chapter, the focus has primarily been on single-input, single-output relationships – in other words, one linking variable vector is supplied for one discipline. However, in both real-world problems, and some single-objective MDO problems, the relationship between disciplines is more complex than this. This is addressed in Chapter 4 as an extension of the work produced in this chapter. Additionally, the benchmark problem is extended to all continuous problems in the ZDT test set; this can be done with ease by modifying the input variables of the objectives and $g(\bullet)$ functions, and introducing linking variables.

Nevertheless, this chapter has provided a good starting point by showing how to construct a simple bi-objective scalable MDO problem, which has allowed an investigation of the impact of two fundamental issues inherent to the MDO architectures from a previously unexplored multi-objective context.

Chapter 5

A scalable test suite for bi-objective multidisciplinary optimization

5.1 Introduction

Most 'real-world' problems in engineering and design consist not only of a large number of disciplines, but also complex linkages between them. In multidisciplinary design optimization (MDO), it is important to model the interactions between subsystems, because the performance of a system is not necessarily defined just by its components, but also by the interactions between those components.

Consider the design of a system with four different disciplines. Figure 5.1 shows how the disciplines within this system may be linked using the structure proposed in Chapter 4.

A realistic system may have many more linkages between the disciplines. Figure 5.2 shows how this might look using the same disciplines as in Figure 5.1.

Like disciplines and design variables, the theme of linkage complexity and scalability has received little attention in the multi-objective multidisciplinary design optimization (MO-MDO) literature.

Figure 5.1: An example of the disciplines in a system with the simple linkages proposed in Chapter 4. The arrows represent how the disciplines are connected.

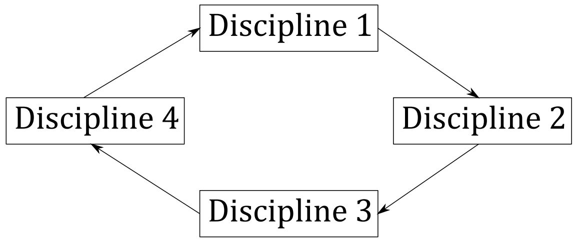
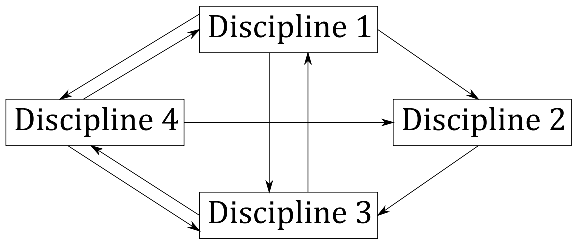


Figure 5.2: An example of a system with more realistic, complex disciplinary linkages.



In Chapter 4, an MDO version of a bi-objective benchmark problem known as ZDT1 was proposed. This problem was then solved using a multidisciplinary feasible (MDF) architecture, using a conventional multi-objective evolutionary optimization algorithm, NSGA-II [111], as the system optimizer, and a Newton-based method as the multidisciplinary analysis (MDA) solver. The present chapter builds on the last chapter and its distinctive contribution is as follows:

1. the approach used to transform the original ZDT1 problem into an MDO variant is extended to the remaining continuous ZDT problems;
2. the way the linking variables are integrated into the optimization problem is improved, in that the deviation of the linking variables from their optimal values is used to perturb the decision variables; and
3. two new topologies for connecting the disciplines via their linking variables are proposed, and it is shown how it is possible to create arbitrary problem structures.

The scope of this chapter encompasses bi-objective MDO problems with both varying number of variables and number of disciplines. The discipline analysis models are mutually interdependent. Although all optimization problems in this chapter contain only continuous variables, discrete variables are also within scope as long as they are supported by the optimization algorithm. Only the MDF architecture has been considered here, but other architectures could be used instead as long they are compatible with the problem formulation.

The remainder of this chapter is organised as follows. Section 5.2 introduces the proposed MO-MDO test suite. Several topologies for connecting the disciplines via linking variables are proposed in Section 5.3. The experimental setup is in Section 5.4, while the obtained experimental results are in Section 5.5. Section 5.6 gives a short summary and discussion of the results presented in this chapter.

5.2 Proposed MO-MDO test suite: ZDT-MDO

The proposed MO-MDO test suite is based on the ZDT benchmark problems, therefore here it is labelled *ZDT-MDO*. Despite the limitations of ZDT as a test set, the original structure of the problems makes them amenable to restructuring into MDO problems in which the original Pareto front is recoverable (which is highly advantageous from an analysis perspective). Here, the five continuous ZDT problems are considered, with ZDT5 omitted because it is binary encoded. For all problems, the first decision variable controls the position across the Pareto front, while the others are called distance decision variables because they control the convergence towards the Pareto front.

The multidisciplinary system contains global variables that are shared between the disciplines, and each discipline has its own set of local variables. The decision variables of the original ZDT problem are partitioned into global and local ones, where the first n_z are global and are represented by the vector $\mathbf{z} = (z_1, \ldots, z_{n_z})^T$. The remaining ones are local variables and are distributed across N disciplines as given by the vector $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$. Each $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n_{x_i}})^T$ contains a total of n_{x_i} local variables at the ith discipline where $i \in \{1, \ldots, N\}$.

The disciplines exchange linking variables to model the interactions of the overall system. These linking variables are the output of an analysis conducted by each discipline that simulates the behaviour of a particular component of the multidisciplinary system. There is a total of n_{y_i} output linking variables at the ith discipline, given by the vector $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,n_{y_i}})^T$, and $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T$ contains the output linking variables of all disciplines. Each discipline may require one or more linking variables from other disciplines to conduct its own disciplinary analysis. To keep track of the linking variable connections in the system consider the following:

1. let n_{p_i} $(1 \leq n_{p_i} < N)$ denote the number of disciplines that provide linking variables to the ith discipline;

2. the indices of the disciplines that provide linking variables to the ith discipline are stored in the set $\mathbf{p}_i = \{p_{i,1}, \ldots, p_{i,n_{p_i}}\}$ where $p_{i,j} \in \{1, \ldots, N\} \setminus \{i\}$ $\forall_{j=1,\ldots,n_{p_i}}$.

For instance, for a hypothetical four-discipline system, if the second and fourth disciplines provide linking variables to the first discipline, then $\mathbf{p}_1 = \{2, 4\}$. The discipline analysis at the ith discipline is to find $\mathbf{y}_i$ that satisfies the following expression:

$$A_{i,i}\mathbf{y}_i + \sum_{j=1}^{n_{p_i}} (A_{i,p_{i,j}}\mathbf{y}_{p_{i,j}}) = -C_i\bar{\mathbf{z}} - D_i\mathbf{x}_i, \tag{5.1}$$

where $\bar{\mathbf{z}} = (z_2, \ldots, z_{n_z})^T$ excludes the first decision variable of the original ZDT problem. The above expression only relies on the decision variables of the distance type, implying that the positional decision variable (z_1) is not included to ensure that there is a single solution to the systems of equations. The matrices in Equation 5.1 are defined as follows:

1. $A_{i,i} \in \mathbb{R}^{n_{y_i} \times n_{y_i}}$, $C_i \in \mathbb{R}^{n_{y_i} \times (n_z - 1)}$, and $D_i \in \mathbb{R}^{n_{y_i} \times n_{x_i}}$ $\forall_{i=1\ldots,N}$,

2. $A_{i,p_{i,j}} \in \mathbb{R}^{n_{y_i} \times n_{y_j}}$ $\forall_{i=1\ldots,N}$ and $\forall_{j=1,\ldots,n_{p_i}}$.

An important aspect of Equation 5.1 is that, depending on how the disciplines are connected, determining the linking variables for one discipline may require knowing the values of the linking variables from the other disciplines. It can become even harder to solve in case there are cyclic connections in the system. The complete set of equations across disciplines can form a full system of equations as given by:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,N} \\ A_{2,1} & A_{2,2} & \dots & A_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N,1} & A_{N,2} & \dots & A_{N,N} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = - \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix} \bar{\mathbf{z}} - \begin{bmatrix} D_1 & 0 & \dots & 0 \\ 0 & D_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D_N \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} \tag{5.2}$$

or equivalently given by:

$$\boldsymbol{A}\mathbf{y} = -\boldsymbol{C}\bar{\mathbf{z}} - \boldsymbol{D}\mathbf{x}. \tag{5.3}$$

To ensure that the full system of equations has a unique solution, $\boldsymbol{A}$ needs to be invertible. Additionally, any column in $\boldsymbol{A}^{-1}\boldsymbol{C}$ or $\boldsymbol{A}^{-1}\boldsymbol{D}$ cannot be all zeros to ensure that there are no redundant design variables. Finding $\mathbf{y_i}$ for all i for the entire system requires the use of numerical techniques, such as Gauss–Seidel and Newton-based methods that are often called multidisciplinary analysis solvers in the MDO literature [24].

The linking variables are incorporated into the optimization problem by penalising the local variables as given by the function:

$$\xi(\mathbf{x}_i, \mathbf{y}_i) = \mathbf{x}_i + \|\mathbf{y}_i - \mathbf{y}_i^*\|_1, \tag{5.4}$$

where $\mathbf{y}_i^*$ are the linking variable optimal values for the ith discipline, and the operator $\|\bullet\|_1$ is the L^1-norm. Let the output of Equation 5.4 be the vector $\hat{\mathbf{x}}_i = (\hat{x}_{i,1}, \dots, \hat{x}_{i,n_{x_i}})^T$, and the function that applies the same transformation to all $\mathbf{x}_i$ is denoted by $\boldsymbol{\xi}(\mathbf{x}, \mathbf{y})$.

The proposed MO-MDO problem formulation based on ZDT1 is given by:

$$
\begin{aligned}
\min \quad & f_1(\mathbf{z}) = z_1 \\
\min \quad & f_2(\mathbf{z}, \boldsymbol{\xi}(\mathbf{x}, \mathbf{y})) = g(\mathbf{z}, \boldsymbol{\xi}(\mathbf{x}, \mathbf{y}))h(\mathbf{z}, \boldsymbol{\xi}(\mathbf{x}, \mathbf{y})) \\
\text{s.t.} \quad & g(\mathbf{z}, \boldsymbol{\xi}(\mathbf{x}, \mathbf{y})) = 1 + \frac{9}{n_v - 1}\left(\sum_{i=2}^{n_z} z_i + \sum_{i=1}^{N}\sum_{j=1}^{n_{x_i}} \hat{x}_{i,j}\right) \\
& h(\mathbf{z}, \boldsymbol{\xi}(\mathbf{x}, \mathbf{y})) = 1 - \sqrt{\frac{f_1(\mathbf{z})}{g(\mathbf{z}, \boldsymbol{\xi}(\mathbf{x}, \mathbf{y}))}}
\end{aligned}
\tag{5.5}
$$

where $n_v = n_z + \sum_{i=1}^{N} n_{x_i}$. For the remaining ZDT problems, $f_1(\mathbf{z}) = z_1$, with the exception of ZDT6 which is $f_1(\mathbf{z}) = 1 - \exp(-4z_1)\sin^6(6\pi z_1)$, while the g and h functions are shown in Table 5.1. For optimality, all decision variables (global and local) with the exception of z_1 have to be zero for the given $g(\cdot)$ functions, unless transformations are applied. This means that Equation 5.2 becomes an homogeneous system of linear equations which is solved when all the $\mathbf{y}_i$ are zero vectors.

The objective functions and corresponding Pareto optimal fronts (POFs) for ZDT2, 3, 4 and 6 are shown in Figure 5.3. The ZDT1 POF can be seen in Figure 4.1.

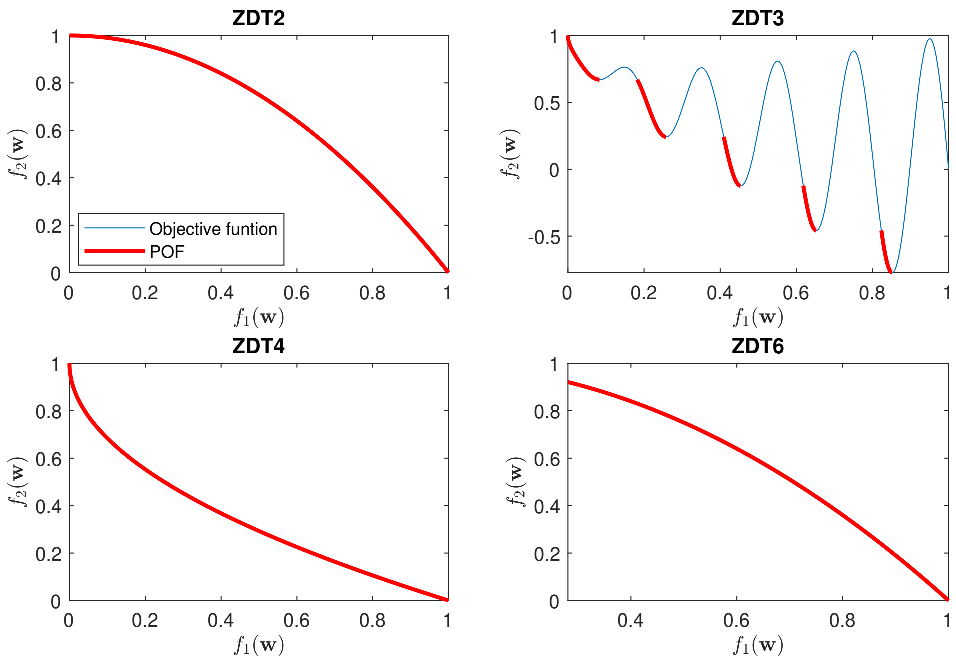


Figure 5.3: The objective functions (in blue) and corresponding POFs (in red) for the continuous ZDT problems, excluding ZDT1.

The benchmarks established in this section can be found in the project's

Table 5.1: The g and h functions of the proposed MO-MDO test suite

	g	h
ZDT2	$1+\frac{9}{n_v-1}\left(\sum_{i=2}^{n_z} z_i+\sum_{i=1}^{N}\sum_{j=1}^{n_{x_i}}\hat{x}_{i,j}\right)$	$1-(f_1/g)^2$
ZDT3	$1+10(n_v-1) + \sum_{i=2}^{n_z}(z_i-10\cos(4\pi z_i)) + \sum_{i=1}^{N}\sum_{j=1}^{n_{x_i}}(\hat{x}_{i,j}-10\cos(4\pi\hat{x}_{i,j}))$	$1-\sqrt{f_1/g}$
ZDT4	$1+\frac{9}{n_v-1}\left(\sum_{i=2}^{n_z} z_i+\sum_{i=1}^{N}\sum_{j=1}^{n_{x_i}}\hat{x}_{i,j}\right)$	$1-\sqrt{f_1/g}$
ZDT6	$1+\frac{9}{n_v-1}\left(\sum_{i=2}^{n_z} z_i+\sum_{i=1}^{N}\sum_{j=1}^{n_{x_i}}\hat{x}_{i,j}\right)^{0.25}$	$1-(f_1/g)^2$

Github repository[1].

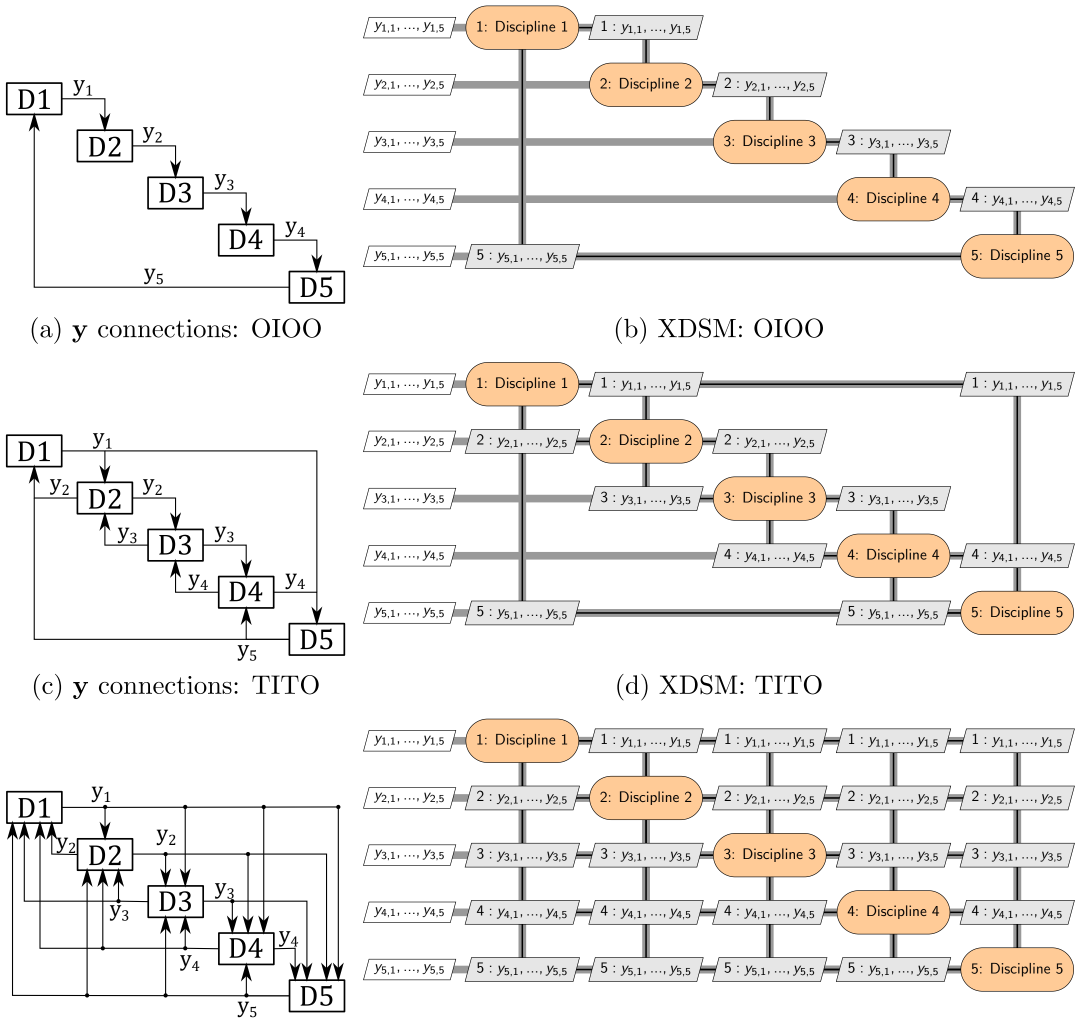
5.3 Defining inter-disciplinary dependencies

The proposed formulation in Equation 5.2 offers the flexibility to connect the disciplines in different ways via linking variables. For instance, for a three-discipline system, in case the second and third disciplines receive linking variables from the first discipline, then $A_{2,1}$ and $A_{3,1}$ have non-zero elements. If there are no more connections between the disciplines (except $A_{i,i}\ \forall_{i=1\ldots N}$ which are set to the identity matrix), then the remaining matrices in $\boldsymbol{A}$ are set to zero. On the other hand, in case the first discipline receives linking variables from either the second or third discipline (implying that $A_{1,2}$ and/or $A_{1,3}$ have non-zero elements), then it can be said that the topology contains cyclic connections.

Figure 5.4a shows a five-discipline system where each discipline is only connected to the next one, and a cyclic connection is created by connecting the last discipline to the first one. The same topology is depicted by an XDSM as shown in Figure 5.4b. Other possible ways of connecting the disciplines are shown in the remaining subfigures in Figure 5.4. The following three topologies are therefore proposed for connecting the disciplines:

1. OIOO: stands for 'one-in-one-out' since each discipline only receives and sends linking variables to a single discipline. A circular topology is adopted, where the first discipline receives linking variables from the last discipline. This is

[1] https://github.com/vj2Sheffield/mdo_zdt



(a) **y** connections: OIOO (b) XDSM: OIOO

(c) **y** connections: TITO (d) XDSM: TITO

(e) **y** connections: AIAO (f) XDSM: AIAO

Figure 5.4: Different topologies showcasing the dependencies between disciplines on a five-discipline system.

given by $\mathbf{p}_1 = \{N\}$ and $\mathbf{p}_i = \{i-1\}\ \forall_{i=2,\ldots,N}$, and the XDSM is shown in Figure 5.4b. This topology was used in Chapter 4.

2. TITO: stands for 'two-in-two-out' since each discipline sends and receives linking variables to two disciplines. This is given by $\mathbf{p}_1 = \{N, i+1\}$, $\mathbf{p}_i = \{i-1, i+1\}$ and $\mathbf{p}_N = \{i-1, 1\}$, and the XDSM is shown in Figure 5.4d.

3. AIAO: stands for 'all-in-all-out' since each discipline sends and receives linking variables to all disciplines. This is given by $\mathbf{p}_i = \{1,\ldots,N\} \setminus \{i\}$, $\forall_{i=1,\ldots,N}$, and the XDSM is shown in Figure 5.4f.

5.4 Experimental setup

The matrices in Equation 5.1 are randomly generated and then row-normalised, as in Chapter 4. The only exception is $A_{i,i}\ \forall_{i=1,\ldots,N}$ which is set to the identity matrix. The number of global variables are set to 10 ($n_z = 10$) and for all disciplines the number of local variables and the size of the linking variables vector is set to 5 (i.e. $n_{x_i} = 5$ and $n_{y_i} = 5\ \forall_{i=1,\ldots,N}$). The lower and upper bounds for the decision variables of all the problems are set to 0 and 1, respectively. The only exception is ZDT4 where the lower bounds are -5 and 5 for all decision variables with the exception of z_1 which takes values in the range $[0, 1]$.

For dealing with the MO-MDO problems, an MDF architecture is adopted, involving a system optimizer and a MDA solver that conducts the disciplinary analysis one discipline at a time. NSGA-II [111] is used as the system optimizer. The crossover and mutation probabilities are set to 90% and $1/n_v$, respectively, while the crossover and mutation index are both set to 20. The number of generations is set to 1000 with a population size of 100. The initial population is randomly initialised. The MDF architecture is provided by the OpenMDAO package in Python [152], and NSGA-II implementation by PyOptSparse [151]. The MDA solver is also provided by OpenMDAO and uses a combination of a nonlinear and linear solver. The nonlinear solver is a Newton method while the linear solver relies on linear algebra techniques such as LU decomposition. The MDA solver runs for a maximum of 1000 iterations. For comparing different problem instances the hypervolume indicator is used. A dimension-sweep algorithm is used for computing the hypervolume, taken from [154]. The reference point used in the hypervolume computations for the first case study (inter-disciplinary relationships) is $\{1.1, 14\}$ for ZDT1, $\{1.1, 13\}$ for ZDT3, $\{1.1, 1620\}$ for ZDT4, and $\{1.1, 17\}$ for ZDT6.

5.5 Experimental results

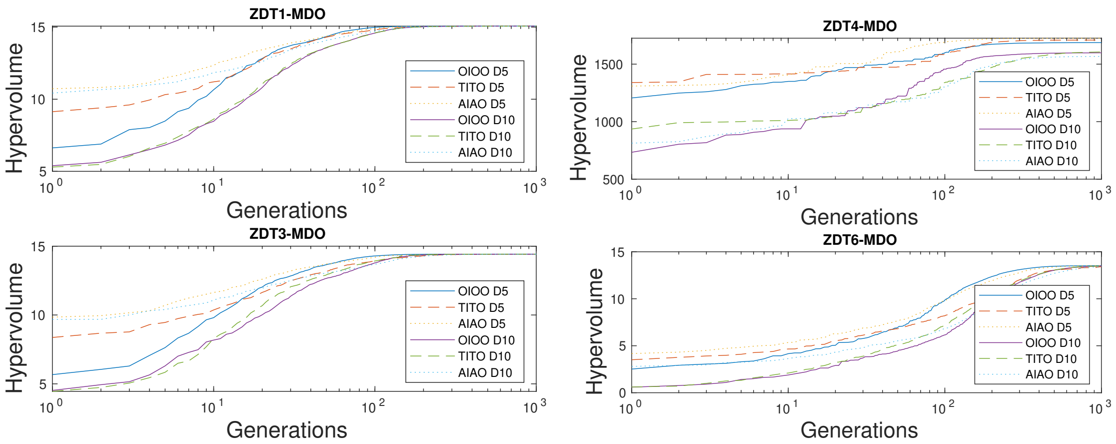
5.5.1 Dependencies between disciplines

In this section the obtained results for the MDO version of ZDT1, ZDT3, ZDT4 and ZDT6 problems using the inter-disciplinary topologies obtained in Section 5.3 are shown, and each subsystem has the same number of local and linking variables. For all cases the MDA solver has run for sufficient number of iterations to guarantee convergence, implying that the correct linking variables were obtained for the given global ($\bar{\mathbf{z}}$) and local variables ($\mathbf{x}$). Therefore our analysis will be mostly focused on the performance of the system optimizer (NSGA-II) in dealing with these problems.

The convergence across generations is captured by the hypervolume metric in Figure 5.5 for five and 10 discipline problems with different linking variable topologies. Figure 5.7 depicts the non-dominated solutions obtained at the end of the optimization run shown alongside the Pareto optimal front (POF). In all plots, the notation D5 and D10 denotes the number of disciplines. Good convergence is achieved for all problems instances involving ZDT1, ZDT3 and ZDT6, although not all solutions are co-located on the POF for ZDT6. ZDT4 shows constant improvement in terms of hypervolume across the generations, but achieves poor convergence overall within the given computational budget.

An increase in the number of disciplines from five to 10 is expected to make the problem more difficult, since it implies an increase in the number of decision variables (30 and 60 decision variables for five and 10 disciplines, respectively). This difficulty is mostly reflected on ZDT4 where the values of f_2 are relatively higher for the 10 discipline case when compared with the five discipline problem. The same trend is captured by the hypervolume for ZDT4, where the five-discipline instances show better convergence when compared with the 10-discipline instances.

The values of the global and local variables across generations are shown in Figure 5.6. Only the five discipline problem is considered, since similar results are obtained for the 10 discipline case. At the end of each NSGA-II generation, the



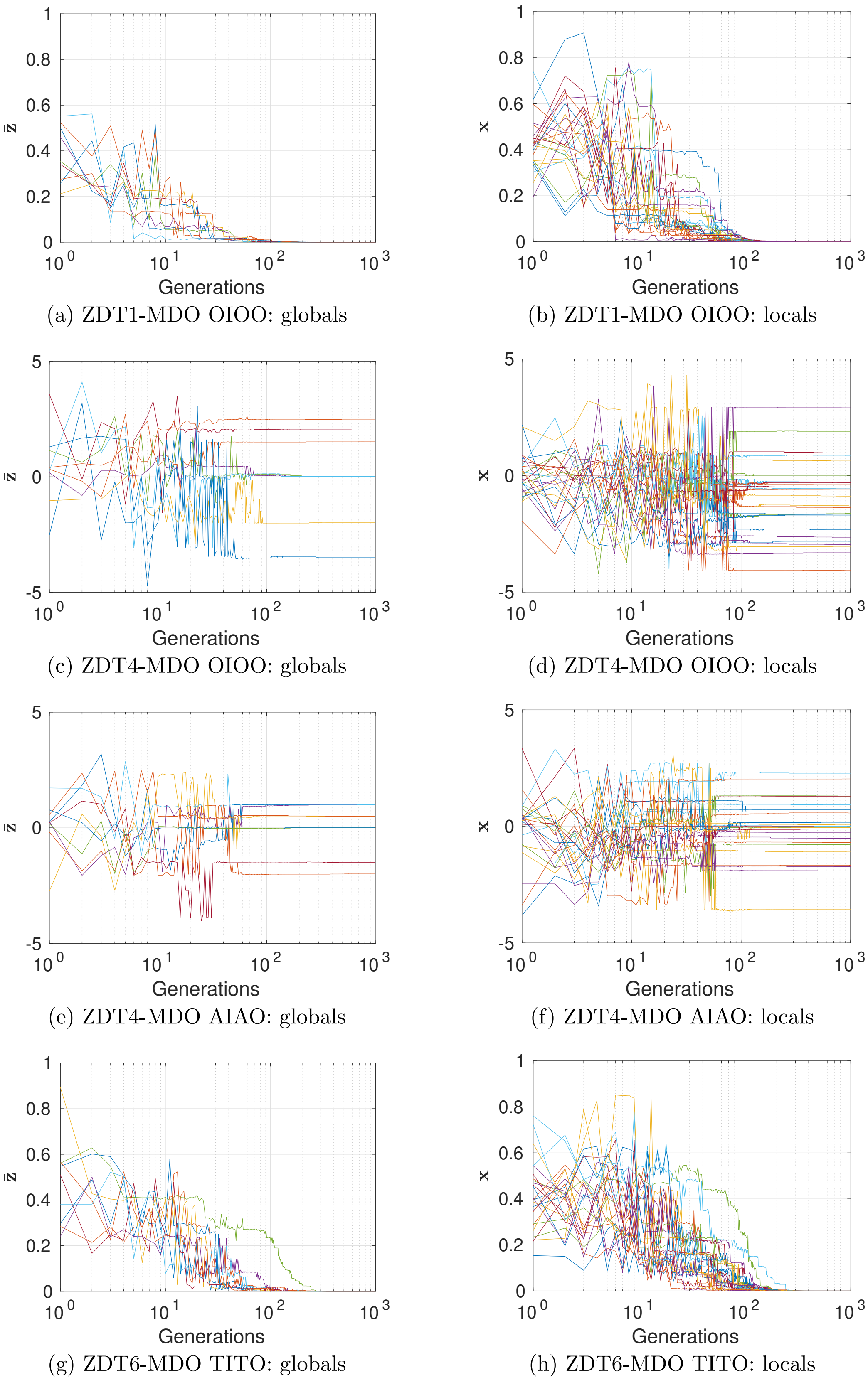
(a) ZDT1-MDO and ZDT3-MDO (b) ZDT4-MDO and ZDT6-MDO

Figure 5.5: Hypervolume across generations.

median of the variable values across the population of solutions is taken. This means that there are 9 lines for the global variables and 25 lines for the local variables in these plots. Figures 5.6a and 5.6b show the global and local variable values, respectively, for ZDT1. Both variables converge towards the optima in less than 200 generations. The same pattern is observed for the other problems with the exception of ZDT4, and it took slightly longer to converge for ZDT6 (Figures 5.6g and 5.6h). The decision variables for ZDT4 become trapped in local optima after a few generations as shown in Figures 5.6c–5.6f. The values of the decision variables for AIAO are relatively close to the optima when compared with OIOO, implying that AIAO achieves better performance when compared with OIOO as shown in Figure 5.7k. Given that the MDA solver has converged in all cases, the differences in performance observed between topologies are likely attributable to the stochasticity of the optimizer at the system level.

5.6 Summary

Linking relationships and system topologies are important aspects of MO-MDO methods and architectures, but these are seldom investigated in the relevant literature. Modifying the structure of one problem and investigating the effects on the outputs of the optimizer is one way of approaching this.



(a) ZDT1-MDO OIOO: globals

(b) ZDT1-MDO OIOO: locals

(c) ZDT4-MDO OIOO: globals

(d) ZDT4-MDO OIOO: locals

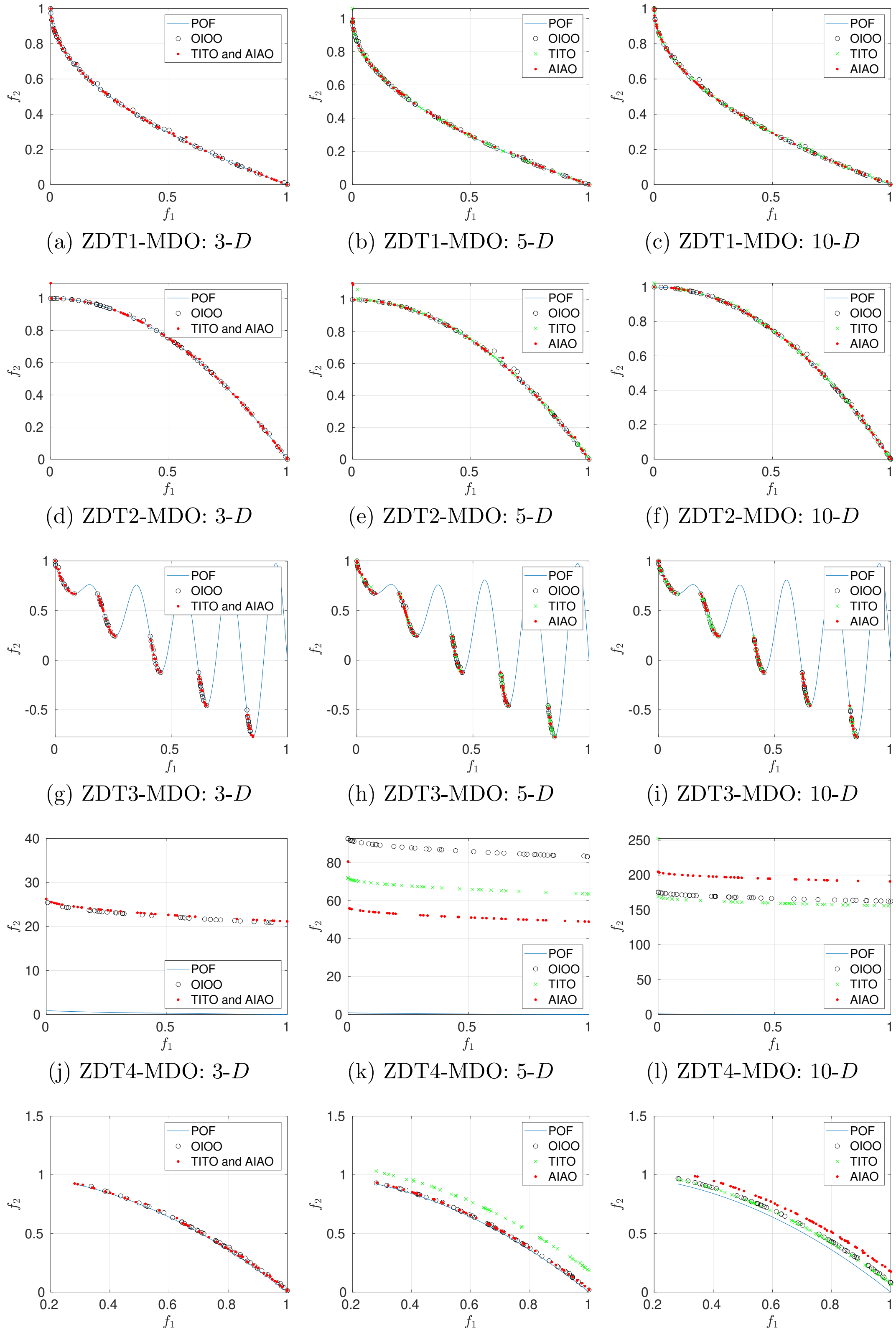
(e) ZDT4-MDO AIAO: globals

(f) ZDT4-MDO AIAO: locals

(g) ZDT6-MDO TITO: globals

(h) ZDT6-MDO TITO: locals

Figure 5.6: Decision variable values obtained across the generations for 5 disciplines.



(a) ZDT1-MDO: 3-D (b) ZDT1-MDO: 5-D (c) ZDT1-MDO: 10-D

(d) ZDT2-MDO: 3-D (e) ZDT2-MDO: 5-D (f) ZDT2-MDO: 10-D

(g) ZDT3-MDO: 3-D (h) ZDT3-MDO: 5-D (i) ZDT3-MDO: 10-D

(j) ZDT4-MDO: 3-D (k) ZDT4-MDO: 5-D (l) ZDT4-MDO: 10-D

(m) ZDT6-MDO: 3-D (n) ZDT6-MDO: 5-D (o) ZDT6-MDO: 10-D

Figure 5.7: Non-dominated solutions at the end of the optimization run.

In this chapter, an MO-MDO test suite has been proposed based on the continuous ZDT problems. The test suite is scalable in the number of disciplines, as well as the number of global and local decision variables. It offers a flexible approach to defining dependencies between the disciplines, allowing for the construction of more complex systems with multiple dependencies between disciplines. This test suite offers the opportunity for researchers and others to develop MDO architectures in combination with multi-objective optimization techniques. The experimental results have shown that for easier ZDT problems, such as ZDT1 and ZDT3, it can be straightforward for an optimizer like NSGA-II and an MDA solver to find a set of solutions with good convergence across the POF. For problems that are harder to solve, such as ZDT4, it may require using an impractical number of generations (beyond 1000) to find a well-converged set of solutions, or a system optimizer more capable of dealing with multimodality in the fitness landscape.

In Section 3.4, a set of suggested criteria for a good benchmark in MO-MDO is detailed. The ways in which the performance of the ZDT1-MDO problem with respect to these criteria is given below:

- Scalability in the number of objectives: This is not met, as the ZDT-MDO problems are based on the bi-objective ZDT problems.
- Scalability in the number of variables: This is met, in both local, linking and global variables.
- Scalability in the number of disciplines: The number of disciplines is scalable in the continuous ZDT-MDO problems.
- Variability in the Pareto front geometry: The ZDT-MDO problems have multiple Pareto front geometries, including convex (ZDT1, ZDT4), concave (ZDT2, ZDT6), mixed (ZDT3), and discontinuous (ZDT3).
- Variability in the complexity of the relationships between disciplines and topologies: In this chapter, three separate topologies have been introduced, meaning the ZDT-MDO problems meet this criterion.
- Extendability to alternate MDO architectures: The problem developed in this

chapter is extendable to other architectures, but this may not always be trivial to achieve.

- Variability in the difficulty of the MDA problem: Only one MDA problem has been used in this chapter, and it is a linear problem. This is a useful consideration for future work.

So far, the work undertaken in Chapters 4 and 5 have been based on the monolithic MDF architecture, where there is one optimizer and the solver resolves the compromise problem between the disciplinary analyses. MDF is useful because it is quick to converge [48] and it is simple to implement. However, other architectures are important to address in MO-MDO research, especially distributed architectures, as these more closely mimic the structure of organisations or companies where decisions are made both by individual departments and the 'executive' level. Collaborative optimization (CO) is one such example. Additionally, constraints are important for consideration in MO-MDO, as all 'real-world' optimization problems contain constraints in some form or another, whether it be on component performance, safety requirements, or financial considerations.

To address these issues, Chapter 6 modifies the test suite introduced in this chapter for use as a multi-objective CO problem. The subsystem optimization problems contain constraints that propagate to the solution sets at the system level, making them easily visible in the objective space.

Chapter 6

A Distributed Multi-Disciplinary Design Optimization Benchmark Test Suite with Constraints and Multiple Conflicting Objectives

6.1 Introduction

In Chapters 4 and 5, an unconstrained bi-objective test suite for MO-MDO was developed using the monolithic MDF architecture. This was scalable in both the number of decision variables and disciplines. MDF is a popular MDO architecture in the literature because the problem it produces is small (since only the local and global variables, as well as the optimization functions are under the control of the optimizer) [24]; and the estimated and real linking variables are always consistent [24]. As a result, it converges quickly and accurately when compared with other architectures [48].

There are other requirements and considerations, such as data privacy and structure, when optimizing complex systems in real-world problems that cannot be met by monolithic architectures such as MDF. The alternative to these are distributed architectures.

In distributed architectures, the optimization problem is decomposed into a set of smaller optimization problems, which produces the same solution when reassembled. The primary motivation for partitioning the problem is to allow different teams or (engineering groups), to work separately on a part of the problem that fits into their own expertise, following a more engineering-like environment.

In addition, most problems involving complex problems will involve constraints. Therefore it is important that constraints are included in the multi-objective CO benchmark problem.

This chapter builds up on the previous two chapters by proposing a distributed version of the test suite based on the collaborative optimization (CO) [156] architecture. Constraints are also introduced at each subproblem, resulting in more realistic MO-MDO problems. The number of constraints is easily scalable via parameter change, which is useful in research and benchmarking contexts, as it allows researchers to understand how increasing number of constraints affects the solutions obtained by the optimizer.

The chapter is structured as follows: Section 6.2 discusses the CO architecture. Section 6.3 introduces the proposed problem and some of the relevant nomenclature. Section 6.4 goes through the proposed method for solving the problem. Section 6.5 gives details on the experimental setup, and Section 6.6 shows the results for case studies one and two. Section 6.7 provides a summary and a conclusion.

6.2 CO architecture

The aim of CO is to allow the optimizers at the subsystem level to work with some degree of independence from other teams. Information-sharing is limited, as each subsystem only has access to its own design variables and the copies of linking variables of other subsystems. This may be useful in real-world applications where data privacy is important. For example, in applications such as decision-making in government, sharing data can be an issue due to security concerns. Similarly, in industrial applications, commercially sensitive information may need to be kept within departments or disciplines to minimise risk of information leaks. Because of this, CO may be a good option for an MDO optimization architecture in such fields.

However, CO has some key difficulties. It has been shown that single-objective CO is generally slow to converge when compared with other architectures [48, 157], especially the monolithic architectures such as multi-disciplinary feasible or all-in-one. Additionally, according to the formulation of CO, increasing the number of design variables means that more copy variables are introduced, meaning that a similar optimization problem executed in a different MDO architecture would contain fewer design variables (and therefore quicker convergence) [157].

Please see Appendix A for the CO architecture formulation.

6.3 Proposed test suite

The proposed test suite builds on the multi-objective MDO test suite in Chapters 4 and 5, based on a monolithic architecture. However, monolithic architectures assume the existence of a single optimization problem, and may not be suitable for dealing with optimization problems that have been partitioned along disciplinary lines (or engineering groups), where each discipline has its own subproblem. The proposed test suite addresses this issue by adopting a distributed architecture based on the CO approach. There are other distributed architectures in the literature [24], but CO has been chosen because the disciplinary subproblems can be independent from each other, allowing teams to control the level of detail shared between them. Another contribution of this chapter is the inclusion of constraints in the problem formulation—a very common property of many engineering problems. Given that the proposed problem formulation is based on the ZDT problems [47], this new test suite is named 'CO-ZDT' in this chapter.

The CO architecture has a system subproblem where it is possible to define performance criteria for the entire system. Each discipline (or subsystem) contains a subproblem with a discipline analysis, and interdependencies can exist between the discipline analysis of the different subsystems. There are three types of design variables: global, local, and linking. Global variables are denoted by $\mathbf{z} = (z_1, \ldots, z_{n_z})^T$ and are accessible to both system and subsystem subproblems. Local variables are distributed across N subsystems, and are only accessible to their subproblems. Let $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n_{x_i}})^T$ which contains n_{x_i} local variables at the ith subsystem where $i \in \{1, \ldots, N\}$. The linking variables are the output of an analysis conducted by each discipline, with the intention of mimicking the behaviour of a particular component in the system. There is a total of n_{y_i} output linking variable at the ith subsystem, given by $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,n_{y_i}})^T$. The disciplinary subproblems are made independent of each other by using copies of the design variables. These are denoted with a hat; for example $\hat{\mathbf{x}}_i$ is the copy of the vector of local variables $\mathbf{x}_i$. The 'copies' of local and linking variables are treated as design variables at the system subproblem, while the 'copies' of the global variables are treated as design variables by the subproblems of the subsystems. The system subproblem is given by:

$$
\begin{aligned}
\text{min} \quad & f_1(\mathbf{z}) = z_1 \\
\text{min} \quad & f_2(\xi(\hat{\mathbf{x}}, \hat{\mathbf{y}}), \mathbf{z}) = g(\xi(\hat{\mathbf{x}}, \hat{\mathbf{y}}), \mathbf{z}) h(\mathbf{z}, \boldsymbol{\xi}(\hat{\mathbf{x}}, \hat{\mathbf{y}})) \\
\text{where} \quad & g(\xi(\hat{\mathbf{x}}, \hat{\mathbf{y}}), \mathbf{z}) = 1 + \frac{9}{N_v - 1} \left(\sum_{i=1}^{N} \sum_{j=1}^{n_{x_i}} \tilde{\hat{x}}_{i,j} + \sum_{j=2}^{n_z} z_j \right) \\
& h(\mathbf{z}, \boldsymbol{\xi}(\hat{\mathbf{x}}, \hat{\mathbf{y}})) = 1 - \sqrt{\frac{f_1(\mathbf{z})}{g(\mathbf{z}, \boldsymbol{\xi}(\hat{\mathbf{x}}, \hat{\mathbf{y}}))}} \\
\text{w.r.t} \quad & \hat{\mathbf{x}}, \hat{\mathbf{y}}, \mathbf{z} \\
\text{s.t.} \quad & 0 \leq \hat{\mathbf{x}}, \hat{\mathbf{y}}, \mathbf{z} \leq 1 \\
& J^* = \sum_{i=1}^{n_z} (z_i - \hat{z}_{1,i})^2 + \sum_{i=1}^{n_z} (z_i - \hat{z}_{2,i})^2 + \\
& \sum_{i=1}^{N} \sum_{j=1}^{n_{x_i}} (\hat{x}_{i,j} - x_{i,j})^2 + \sum_{i=1}^{N} \sum_{j=1}^{n_{y_i}} (\hat{y}_{i,j} - y_{i,j})^2 \leq \varepsilon
\end{aligned}
\tag{6.1}
$$

where the linking variables have been incorporated into the subproblem via the following function:

$$\xi(\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i) = \hat{\mathbf{x}}_i + ||\hat{\mathbf{y}}_i - \mathbf{y}_i^*||_1. \tag{6.2}$$

In Equation 6.2 the decision variables are penalised by the deviation between the linking variables and their optimal values ($\mathbf{y}_i^*$). The operator $|| \bullet ||_1$ is the L^1-norm. Let the output of Equation 6.2 be denoted by the vector $\tilde{\hat{x}}_i = (\tilde{\hat{x}}_{i,1}, \ldots, \tilde{\hat{x}}_{i,n_{x_i}})^T$. The function J^* is a consistency constraint that quantifies the difference between the values of the decision variables (and linking variables as well) that are kept by the system and subsystems subproblems. Given that the consistency constraint can be difficult to satisfy (implying that the deviation between system and subsystems is zero), the constraint is considered to be satisfied when the deviation is below a small number ε (bigger than zero). The subproblem at the kth subsystem where $k \in \{1, \ldots, N\}$ is given by:

$$\begin{aligned} \min\ J_k(\mathbf{x_k}, \mathbf{y_k}, \hat{\mathbf{z}}_\mathbf{k}) = &\sum_{i=1}^{n_z} (z_i - \hat{z}_{k,i})^2 + \sum_{i=1}^{n_{x_k}} (\hat{x}_{k,i} - x_{k,i})^2 + \\ &\sum_{i=1}^{n_{y_k}} (\hat{y}_{k,i} - y_{k,i})^2 \end{aligned} \tag{6.3}$$

$$\text{w.r.t } \mathbf{x_k}, \mathbf{y_k}, \hat{\mathbf{z}}_\mathbf{k}$$

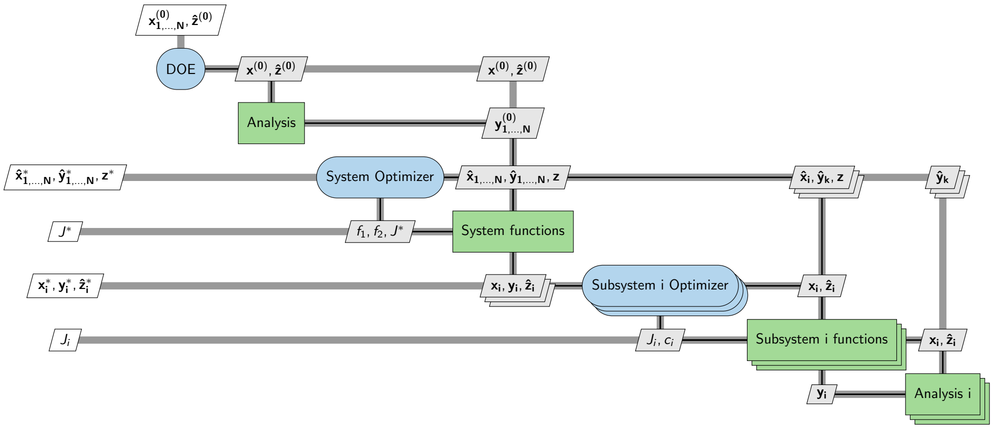
$$\text{s.t. } c_k \equiv -|\alpha_k(\hat{z}_1 - \beta_k)| + 1 \leq 0 \text{ and } 0 \leq \mathbf{x_k}, \mathbf{y_k}, \hat{\mathbf{z}}_\mathbf{k} \leq 1$$

The objective function resembles the consistency constraint found in the system subproblem (Equation 6.2), and its aim is to reduce the inconsistency with respect to the kth subsystem. The constraint function c_k generates infeasible regions in f_1, where α determines the width of the regions and β specifies their lateral placement. Based on the experimental results it is recommended that α is set to 0.02. The value of β depends on the number of subsystems and should be set in a way that produces a number of evenly spaced feasible regions across the objective space. The values of the linking variables are determined at the subsystem level and require solving the following system of equations:

$$\boldsymbol{A}\mathbf{y} = -\boldsymbol{C}\hat{\hat{\mathbf{z}}} - \boldsymbol{D}\mathbf{x}, \tag{6.4}$$

where $\hat{\hat{\mathbf{z}}} = (\hat{\mathbf{z}}_2, \ldots \hat{\mathbf{z}}_{n_z})^T$ excludes the first shared variable used in f_1, $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T$ contains all linking variables, and all decision variables are in $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$. The matrices $\boldsymbol{A}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ specify the couplings between the subsystems; more details about these matrices can be found in [58]. To solve the system of equations

Figure 6.1: The XDSM diagram for the multi-objective collaborative optimization strategy used in this chapter.



in Equation 6.4, a multidisciplinary analysis solver from the MDO literature can be used (e.g. Gauss–Seidel or Newton-based methods [24]).

6.4 Proposed method and termination criteria

The method used in this chapter builds upon the single-objective approach for CO, but is different from some of the existing multi-objective CO strategies such as COSMOS [9] in that each member of the population is optimized separately.

1. The variables $\mathbf{x_i}^{(0)}$ and $\hat{\mathbf{z}}^{(0)}$ are initialised using a Latin hypercube design of experiments. This is of size $nPop$, the size of the population. The matrices B_i, C_i and D_i, used in the subsystem analyses, are also initialised and row-normalised.

2. The subsystem analyses are run once, using the values of $\mathbf{x_i}^{(0)}$ and $\hat{\mathbf{z}}^{(0)}$ established in Step (1). These return the values of the initial linking variables $\mathbf{y_i}^{(0)}$.

3. The initial global, local and linking variables are sent to the consistency constraint J^* in the system-level optimization problem shown in 6.3. The system-level optimizer then uses its own design variables $\hat{\mathbf{x}}_\mathbf{i}$, $\hat{\mathbf{y}}_\mathbf{i}$ and $\mathbf{z}$ to find a set of solutions that satisfy the consistency constraint and are close to optimal in objective functions f_1 and f_2. At this stage in the optimization, ε is a large

number, for example 10, to guarantee a set of solutions are found.

4. The main optimization loop takes place:

   (a) The subsystems receive the variables $\hat{\mathbf{x}}_\mathbf{i}$, $\hat{\mathbf{y}}_\mathbf{i}$ and $\mathbf{z}$, obtained by the initial optimization in Step (3). They perform a single-objective optimization using the variables $\mathbf{x_i}$ and $\hat{\mathbf{z}}$ with the aim of minimising the consistency objective J_i and satisfying the subsystem constraints c_i. The linking variables $\mathbf{y_i}$ are obtained by running the subsystem analyses shown in Equation 6.3, and are used in the consistency objectives.

   (b) When optimization at the subsystem level has been terminated, the system-level problem receives the variables $\mathbf{x_i}$, $\mathbf{y_i}$ and $\hat{\mathbf{z}}$ from the subsystems. The system then runs the optimization again, similarly to that in Step (3), but ε is set to a much lower value – in this chapter, between 10^{-1} and 10^{-7}. Subject to the consistency constraints, the system solves the problem with variables $\hat{\mathbf{x}}_\mathbf{i}$, $\hat{\mathbf{y}}_\mathbf{i}$ and $\mathbf{z}$.

   (c) The termination criteria, described below, are assessed. If the termination criteria are satisfied, or all the permitted supervisor iterations have been expended, the optimization terminates. If the termination criteria are not satisfied, and the number of completed supervisor iterations is lower than the permitted number, the optimization loop repeats from Step (4)(a).

5. When the optimization process has been terminated, the results are stored for further analysis.

The above method is also described in the extended design structure matrix (XDSM) in Figure 6.1. See [63] for more information on XDSM diagrams.

There are three termination criteria:

1. The constraint on the shape variable expressed in 6.3 is satisfied.

2. The constraints on the linking variables expressed in 6.3 are satisfied to within 10^{-15}.

3. The consistency constraint in the system-level problem is satisfied by all solu-

tions (i.e. the maximum J^* must be equal to or less than ε).

There are a maximum of 10 permitted supervisor (system-subsystem) iterations for the sake of reducing experimental time while allowing the optimizers a suitable number of function evaluations to find a permissible result. When all of the above termination criteria are met, the optimization process is terminated. If all of the above criteria are not met within the 10 supervisor iterations, the optimization is terminated and the results are stored.

The code used to conduct the method described in this section are available in the associated GitHub repository[1].

6.5 Experimental setup

The following experiments were undertaken in Python, making use of the PyMoo package [158] for the system-level optimizer and SciPy [159] for the subsystem-level optimizers. The system-level optimizer is NSGA-II [111]. For each run of NSGA-II, a population of 50 is used with a number of generations that is varied between 2000 and 10,000 in the experiments. The subsystem-level optimizers are both SLSQP with a maximum of 200 iterations and a function tolerance of 10^{-5}. This is to prevent excessive optimization time. The optimal linking variable values y^* from Equation 6.3 is a vector of zeros of length n_{y_k} for each subsystem k. Each optimization is repeated 11 times for statistical significance. The hypervolumes of each of the repetitions at the final supervisor iteration are measured, and the median repetition is used for analysis.

The constraints are spread equally between 0 and 1 in the z_1 variable. For example, for a 2-discipline system, there would be 2 equally spaced constraints, and in a 5-discipline system, there would be 5 equally spaced constraints.

The first experiment involves varying ε between 10^{-1} and 10^{-7}. Two subsystem problems will be used, and the number of generations will not be allowed to vary throughout. Each of the ZDT problems except 3 and 5 are used.

[1] https://github.com/vj2Sheffield/CO_ZDT_benchmarks

The second experiment looks at the evolution of the design variables and the consistency constraint at the system level for CO-ZDT1, 2, 4 and 6 for two, three, five and ten subsystems. ε is set to 10^{-7}.

The third experiment investigates the outcomes of the optimization when 10,000 generations are used in a problem with 2 subsystems. Once again, ε is varied between 10^{-1} and 10^{-7}.

The fourth study extends the approach in the first experiment to three subsystems. 2000 generations are used and ε is varied between 10^{-1} and 10^{-7}.

6.6 Results

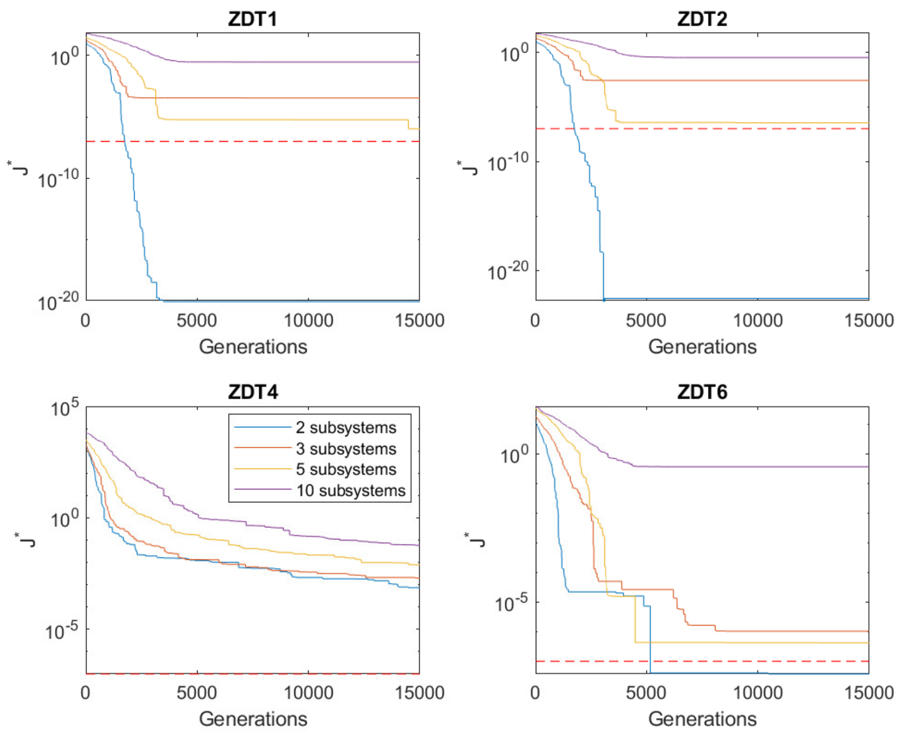
6.6.1 Convergence in a Single Supervisor Run of Multi-objective CO

The speed of convergence is of particular interest in collaborative optimization. In case study one, the number of allotted generations is set to 15,000 for one design point (15,000 function evaluations) to allow the optimizer to reach an acceptable value. This run takes place after the initial system-level optimization and a single subsystem-level optimization – in other words, this is after one supervisor iteration. The experiments were also undertaken with two, three, five and ten subsystems to demonstrate the effects of increasing the number of subsystems on convergence speed.

Figure 6.2 shows the value of the consistency constraint J^* over the 15,000 generations for CO-ZDT1, 2, 4 and 6 and two, three, five and ten subsystems. When the consistency constraint value crosses the red dashed line, it indicates that the constraint is satisfied, and in a run of the multiobjective CO method outlined in Section 6.4, termination criterion (1) would be satisfied.

It can be seen that the only cases where the consistency constraint is satisfied is in CO-ZDT1, 2 and 6 for two subsystems. CO-ZDT1 and 2 show a similar evolution of constraint values, with each of the different subsystem numbers reaching

Figure 6.2: The values of the consistency constraints J^* with a budget of 15,000 generations and for 2, 3, 5 and 10 subsystems. The red dashed lines indicate the placement of the constraint $\varepsilon = 10^{-7}$.



similar final values. The ten-subsystem problems show the slowest reduction in the consistency constraint, settling at just below 1. The CO-ZDT4 and 6 problems show similar results, with 10 subsystems demonstrating the most shallow curve. Additionally in both the CO-ZDT4 and 6 problems, the final value of the consistency constraint is lowest in two subsystems, followed by three, five and ten subsystems.

Figure 6.3 shows the convergence of the global (excluding z_1), copy local and linking variables for CO-ZDT1, 2, 4 and 6 problems with two subsystems. The variables shown in CO-ZDT4 are absolutes to demonstrate the movement of the variables at very small values. Like in Figure 6.2, ZDT1 and 2 demonstrate similar convergence patterns, reaching their final values before the 4000th generation. In CO-ZDT4, the variables decrease in the first 5000 generations, then stagnate between 10^{-4} and 10^{-2} until the 15,000th generation. In CO-ZDT6, the variables drop in the first 2000 generations, stall at approximately 10^{-13}, and then decrease again just before the 10,000th generation.

Figure 6.3: The convergence of the global, copy local and copy linking variables in one run of the optimizer with $\varepsilon = 10^{-7}$, excluding $\mathbf{z_1}$, and a budget of 15,000 generations and two subsystems. The variables in CO-ZDT4 are absolutes to demonstrate the convergence movements at smaller values.

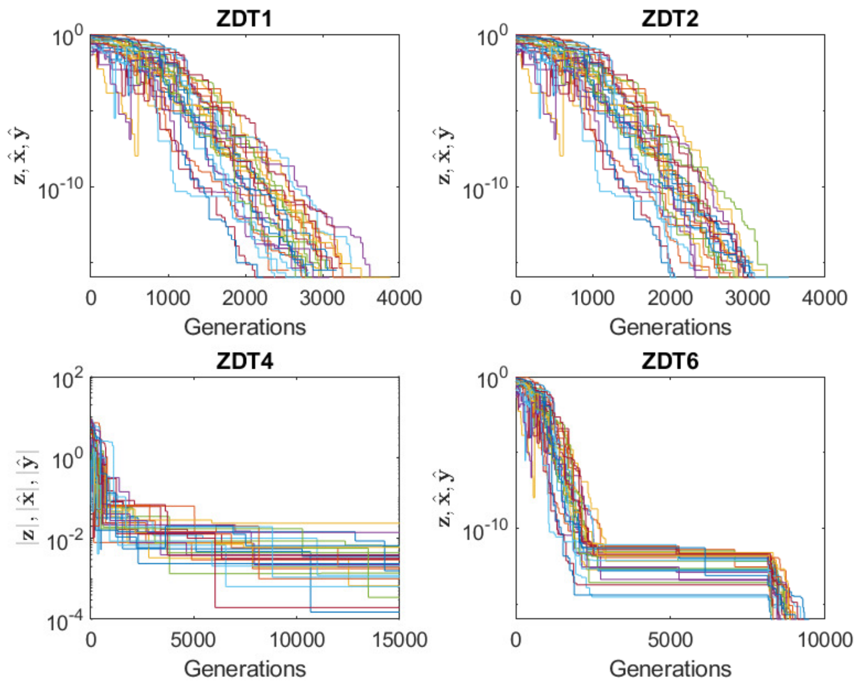


Figure 6.4: The convergence of the global, copy local and copy linking variables in one run of the optimizer with $\varepsilon = 10^{-7}$, excluding $\mathbf{z_1}$, and a budget of 15,000 generations and ten subsystems.

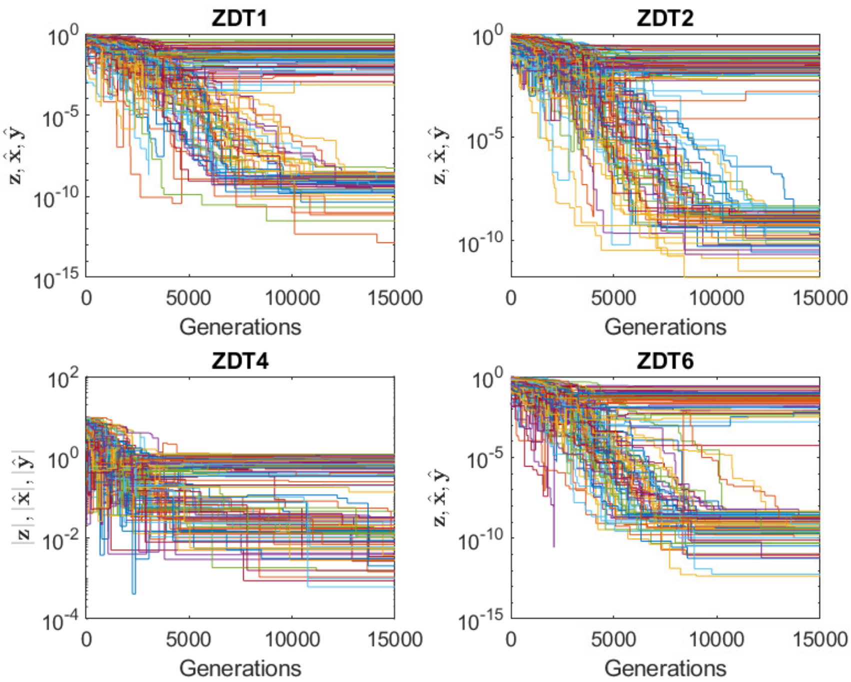


Figure 6.4 shows the convergence of all variables for the problem where there are ten subsystems. In these problems, there are 110 variables in total: 10 global variables, 50 local copy variables, and 50 linking copy variables. It can be seen that some of the variables decrease substantially, especially in CO-ZDT1, 2 and 6. However, some of them also remain between 10^{-1} and 10^{-5} and stagnate.

6.6.2 System Runs with 2000 Generations

In case study two, CO-ZDT1, 2, 4 and 6 were studied using the multi-objective CO method described in Section 6.4 using a set number of generations, population, and maximum supervisor iterations.

Figure 6.5 shows the mean number of supervisor iterations used before termination for each of the CO-ZDT problems and each ε value. It can be seen that, for all CO-ZDT except CO-ZDT4, the number of supervisor iterations at termination increase steadily until $\varepsilon = 10^{-4}$ and then jumps for smaller values in a shape

Figure 6.5: The mean number of supervisor iterations completed at termination.

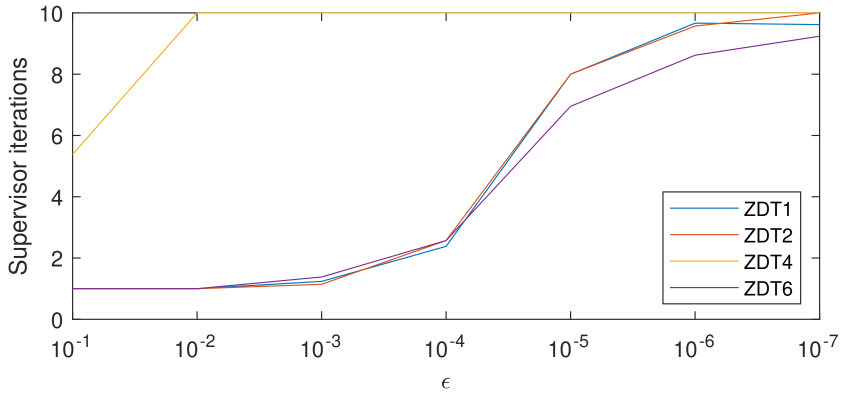
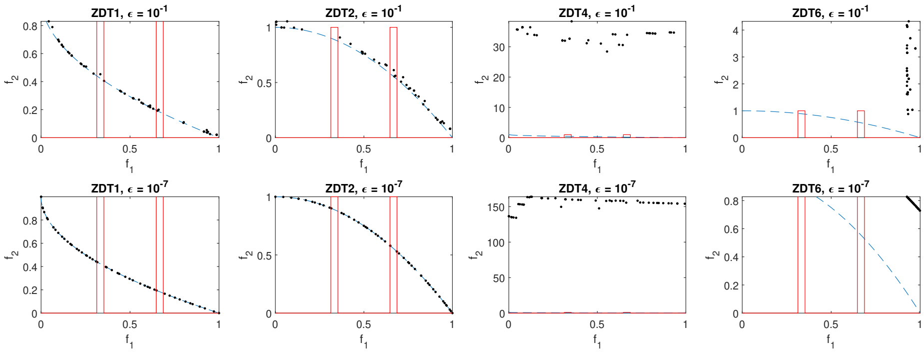


Figure 6.6: Nondominated solutions at termination for $\varepsilon = 10^{-1}$ and 10^{-7} for CO-ZDT1, 2, 4 and 6, in a two-subsystem problem.

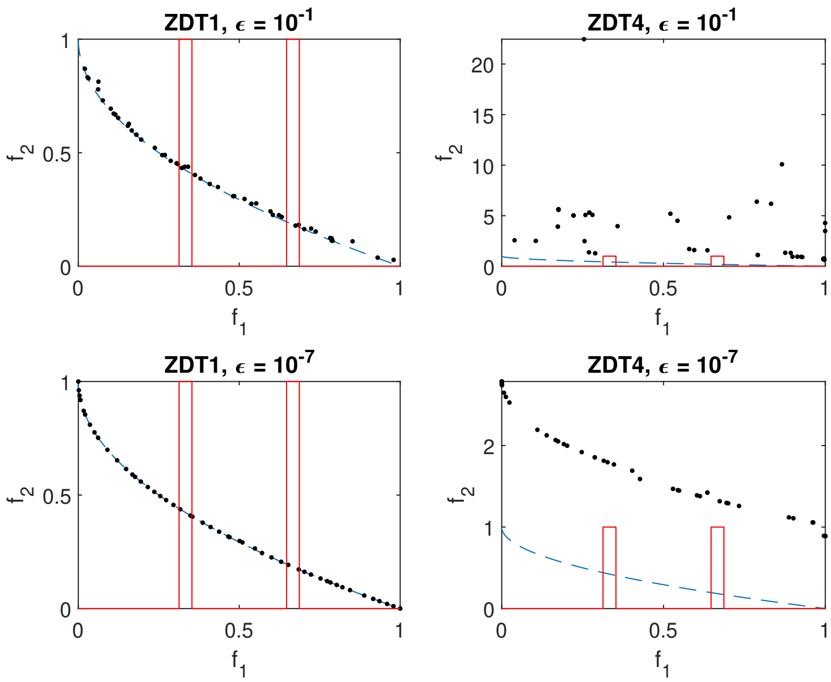


resembling a sigmoid curve. In CO-ZDT4, the mean number of iterations starts at just under 6, then jumps to 10 for the remaining ε values.

Figure 6.6 shows the nondominated solutions at termination for each of the CO-ZDT problems for $\varepsilon = 10^{-1}$ and 10^{-7}. Firstly, the solutions for CO-ZDT1 and 2 are easily recognisable as they adhere closely to the Pareto front, while CO-ZDT4 and 6 do not, with CO-ZDT4 displaying large f_2 values and CO-ZDT6 having solutions limited to the values of f_1 larger than 0.9.

In terms of constraint violation, the problems where ε is very low shows better constraint adherence than in those where ε is larger. Additionally, diversity along the nondominated solutions is greater in problems where ε is small than in those where it is large. Interestingly, a smaller ε appears to lead to f_2 values in CO-ZDT4 that are further away from the Pareto front than larger values - $\varepsilon = 10^{-1}$

Figure 6.7: Nondominated solutions at termination for $\varepsilon = 10^{-1}$ and 10^{-7} for CO-ZDT1 and 4 in a two-subsystem problem with 10,000 generations at the system level.



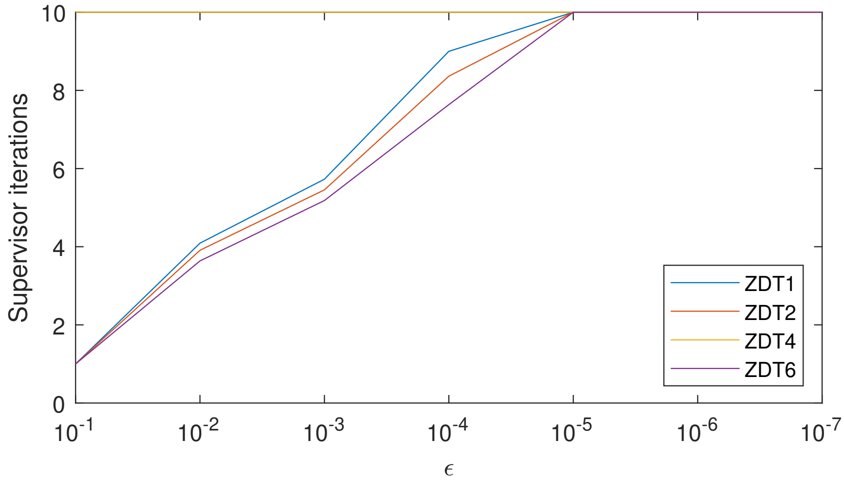
tops out at just under 40, while $\varepsilon = 10^{-7}$ reaches over 150. The opposite effect is found in CO-ZDT6, where f_2 becomes smaller with a smaller ε; the largest f_2 solution in $\varepsilon = 10^{-1}$ is over 4, while the largest in $\varepsilon = 10^{-7}$ is just over 0.8.

6.6.3 System Runs with 10,000 Generations

In case study three, the number of generations was increased to 10,000, and the number of subsystems in the problem was maintained at two.

Figure 6.7 shows the nondominated solutions for CO-ZDT1 and 4 in a two-subsystem problem, $\varepsilon = 10^{-1}$ and 10^{-7} and 10,000 generations. It can be seen that the solutions at $\varepsilon = 10^{-1}$ are much less evenly spread across the objective space when compared with $\varepsilon = 10^{-7}$. In CO-ZDT4 at $\varepsilon = 10^{-1}$, the shape of the Pareto front is not recovered. However, at $\varepsilon = 10^{-7}$ for CO-ZDT4, the shape is recovered, and the nondominated points are much closer to the Pareto front in f_2.

Figure 6.8: The mean number of supervisor iterations completed at termination for three subsystems.



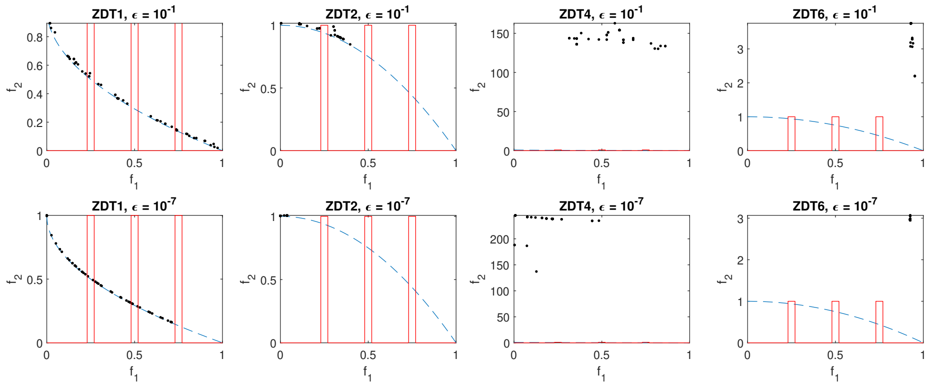
6.6.4 System Runs with 2000 Generations and 3 Subsystems

In case study four, the number of disciplines was increased to three and the number of generations was set at 2000, as in experiment one.

Figure 6.8 shows the mean number of supervisor iterations at termination for each of the CO-ZDT problems studied in this chapter. CO-ZDT1, 2 and 6 trend upwards from $\varepsilon = 10^{-1}$ to 10^{-5}, and then reach 10 supervisor iterations. CO-ZDT4 begins with an average of 10 supervisor iterations, where it remains for all of the ε values. Like in case study one, the results indicate that for all the CO-ZDT problems studied in this chapter, the number of supervisor iterations needed for termination increases with smaller values of ε.

Figure 6.9 shows the nondominated solutions for the three-discipline problem using the same format as in Figure 6.6. The constraints on z_1 are shown by the red lines while the black dots show the nondominated solutions. For CO-ZDT1, it can be seen that the nondominated solutions for both $\varepsilon = 10^{-1}$ and 10^{-7} are close to the Pareto front, but in $\varepsilon = 10^{-1}$ there are greater constraint violations. Additionally, in $\varepsilon = 10^{-7}$, the nondominated solutions fail to cover the whole range of possible z_1, meaning the solutions do not extend past approximately 0.73. The spread gets worse

Figure 6.9: Nondominated solutions at termination for $\varepsilon = 10^{-1}$ and 10^{-7} for CO-ZDT1, 2, 4 and 6, in a three-subsystem problem.



with CO-ZDT2, especially at the smaller value of $\varepsilon = 10^{-7}$, where the solutions only extend to 0.04, and 0.4 for 10^{-1}. In CO-ZDT4, the solutions are more concentrated in the middle of f_1 for $\varepsilon = 10^{-1}$, but is less resemblant of the shape of the Pareto front than in 10^{-7}. Like in case study 1, the nondominated solutions are further away from the Pareto front in $\varepsilon = 10^{-7}$. Again, similarly to in case study 1, the solutions for CO-ZDT6 are limited to the larger values of f_1 in both $\varepsilon = 10^{-1}$ and 10^{-7}.

6.7 Summary and conclusion

6.7.1 Summary

In this chapter, a set of four scalable benchmark problems have been proposed for multi-objective collaborative optimization based on the ZDT test suite. These scalable benchmark problems have varying numbers of subsystems (between two and three) and each subsystem problem includes one constraint, which is reflected in the solution set of the system problem. The relaxation coefficient applied to the system-level consistency constraint is varied between 10^{-1} and 10^{-7} and the number of supervisor iterations taken for termination criteria satisfaction is recorded. The results of running the optimization with the NSGA-II optimizer at the system level and the SLSQP optimizer at the subsystem level have been investigated, visualised and analysed.

It is found that that the problems tend to be slow to converge, and benefit from the use of a larger budget of function evaluations at the system level. In addition, the number of subsystems also tends to decrease the speed of convergence, and that in CO-ZDT4, the optimizers tended to find a solution that satisfied the consistency constraints much more slowly than in other CO-ZDT problems.

On the Speed of Convergence in CO

It is well-established that CO takes a long time to converge on a solution. Tedford and Martins show that, out of several monolithic and distributed architectures in single-objective problems, CO often demonstrates the slowest convergence [48]. Alexandrov and Lewis note the poor efficiency in CO due to the high autonomy afforded to the subsystems [157] (and again in [160]), especially in problems with large numbers of design variables (often caused by the structure of CO itself). Cormier et al. state that the slow convergence associated with CO had a detrimental effect on their design of a reusable launch vehicle [161].

Given the above, and the results in Section 6.6, it is found that the results are in line with the existing literature on CO. This architecture produces more complex problems than 'vanilla' ZDT and therefore requires more computational resources.

6.7.2 Conclusion and future work

Establishing a set of scalable, constrained benchmarks in the area of MO-MDO opens up a wide range of research approaches for specialists in other fields, such as multi-criterion optimization, multi-disciplinary design and classical optimization.

Multi-objective MDO is an emerging field and further work in this area could take many different directions. Firstly, and most importantly, it should be stated (as before) that problems closely related to the ZDT test suite do not have attributes that are desirable for a bi-objective BBOB problem—lacking scalability in objectives, relying on the leading variable to determine the shape of the nondominated solution front, and lack of non-separability in objective functions are hurdles that need to be overcome in a good MO-MDO benchmark problem. The benchmark test set proposed in this chapter inherits the aforementioned problems.

Secondly, the number of supervisor iterations could be increased, with an analysis on how long it takes the optimization to reach a satisfactory solution.

Indicator-based constraints could be introduced as a requirement for termination, in addition to the termination criteria set out in Section 6.4. An example of this would be adding a termination criterion stating that a certain hypervolume must be achieved before the optimization process is allowed to terminate.

Additionally, future work should allow for some variation of the optimization problems by linking the subsystems together in different ways. In this work, each subsystem has contributed linking variables to and received linking variables from only one other discipline. In further work, subsystems should receive and contribute linking variables to and from multiple subsystems. This complicates the problem, and is more realistic to real-world problems.

In Chapter 7, constraints are used within a 'real-world' problem similar to those faced by devolved authorities in the UK. This problem is in public health policy making. Both the objectives and the disciplinary analyses in this problem are typically more complex than those presented by the modified ZDT test set presented in the last three chapters. This addresses one of the issues presented by using the modified ZDT test set. Additionally, the variation of subsystem linkages mentioned before is addressed in the next chapter, as the relationships between the subsystems become more complex.

Chapter 7

Applying the MO-MDO framework to a public health policy problem in devolved government

7.1 Introduction

In Chapters 4, 5 and 6 of this thesis, the research focus was centred on academic multi-objective multidisciplinary design optimization (MO-MDO) problems, by adapting the ZDT test suite to the multidisciplinary feasible (MDF) and collaborative optimization (CO) MDO architectures. In all these previous chapters, it was found that increasing the number of disciplines slows convergence of the optimiser at a greater rate than correspondingly increasing the number of variables in the non-MDO ZDT1 problem.

In Chapter 7, some of the key aspects of these chapters will be drawn upon to develop an applied multi-objective multidisciplinary design optimization (MO-MDO) problem. An MDF architecture is chosen for adoption, as the results of Chapter 6 found that the CO architecture struggled with larger numbers of disciplines when compared with the results of the ZDT-MDO problems in the MDF architecture. The number of disciplines that an MO-MDO framework is capable of handling is important when considering problems in local or regional authorities, because these decision problems are often organised across a large number of disciplines [162].

Additionally, the GMCA problem in this chapter shares some aspects with the ZDT1-MDO problem developed in Chapter 4. This is that they both have convex and continuous Pareto fronts (although this is only an approximate Pareto front for the GMCA problem, since the true front is unknown). Additionally, there are no multimodal, discontinuous or mixed elements in the GMCA problem.

In this chapter, the MDF architecture will be used to handle an MO-MDO problem adapted from a real-world setting – a combined local authority. Local authorities (LAs) are decision-making bodies that hold a range of powers, including choosing how to allocate funding across the local area, both geographically and across different services, such as housing and waste collection. LAs (or models of LAs) are excellent choices for undertaking research into MO-MDO, because LAs can typically be split up into different 'subsystems', including geographic subsystems and service subsystems. The interactions between these subsystems can be computationally complex, or there may be many connections between subsystems leading to a large number of calculations happening in one MO-MDO run. In this chapter, a static model is used to model a combined local authority, and the MDF architecture is used to solve an optimization problem centred on health inequalities within the LA problem.

In this chapter, a background to the problem and key concepts is introduced in Section 7.2. The systems map and model are introduced in Section 7.3. Section 7.4 outlines the optimization problem in the MDF architecture, as well as the disciplinary analyses. The decision modelling used before the optimization process is presented in Section 7.5. The setup used in the optimization is shown in Section 7.6, and Section 7.7 shows the results for the problem, including the baseline for comparison. Section 7.8 gives a summary and conclusions of the chapter.

7.2 Background

7.2.1 Census geography

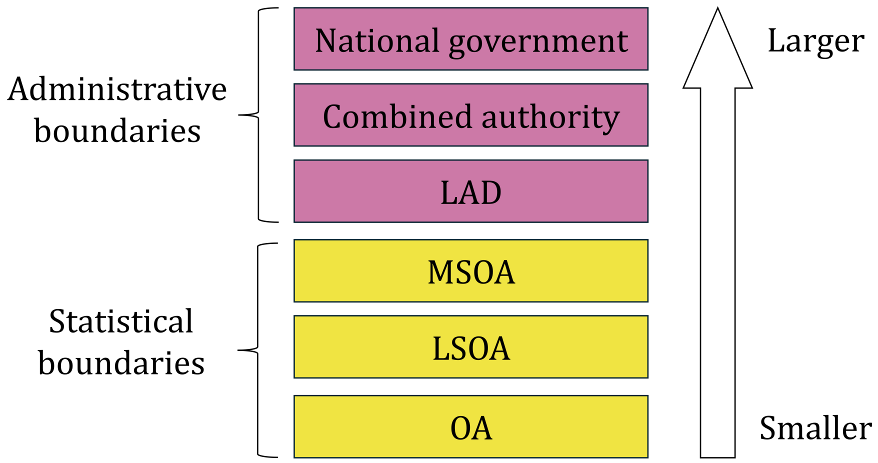
In the United Kingdom, when a census is taken, the data are organised into geographical areas. The smallest of these areas is known as an output area (sometimes referred to as an OA) and contains no more than 625 people or 250 households [163]. Super output areas are made up of groups of OAs and represent larger populations. Lower super output areas (LSOAs) contain between 400 and 1200 households or 1000 to 3000 people, and middle layer super output areas (MSOAs) contain between 2000 to 6000 households or 5000 to 15 000 people [163, 164]. Local authority districts (LADs) are different from OAs and SOAs in that they exist to define the boundaries of the areas that governing bodies are responsible for. For example, the LAD Oldham is governed by Oldham Metropolitan Borough Council. Additionally, LADs are much larger than OAs and SOAs. OAs and SOAs are mainly used for statistical purposes, while LADs, combined authorities and national boundaries are used for administrative purposes. Combined authorities are made up of multiple LADs, such as in the case of Greater Manchester Combined Authority (GMCA) which has ten LADs to one combined authority.

Greater Manchester Combined Authority GMCA is a combined authority that comprises ten LADs near and around Manchester. Representatives from the ten LADs run the combined authority jointly.

7.2.2 Clustering LADs in GMCA

Greater Manchester comprises ten local authority districts, shown in Table 7.1, along with their abbreviations. Each LAD is assigned an index number for ease of use. These LADs are also assigned a cluster based on their index of multiple deprivation (IMD).

Figure 7.1: The statistical and administrative hierarchies of OAs, SOAs, LADs and higher administrative boundaries such as combined authorities and national government



The IMD in England is an indicator of relative deprivation. Its calculation uses seven 'domains' of deprivation: income; employment; health deprivation and disability; education, skills and training; crime; barriers to housing and service; and living environment [165].

The IMD is calculated for every LSOA in England. These are then ranked nationwide and the deciles of the ranked LSOAs are used to describe the relative deprivation. A higher decile means more relative deprivation – for example, an LSOA with an IMD in the 10th decile is in the top 10% of most deprived LSOAs in England.

For the GMCA MO-MDO problem, all LAs in England are ranked according to the average IMD ranking for LSOAs in the LA. These LAs are then ranked and split into four equal clusters (quartiles). The same principle applies for LADs as it does for LSOAs – the higher the quartile, the greater the relative deprivation.

The map in Figure 7.2 shows which cluster each LAD is in. As an example, Trafford (cluster two) is less deprived than Wigan (cluster three), which is in turn less deprived than Oldham (cluster four).

Table 7.1 shows the cluster each LAD is in, as well as the abbreviations and assigned index numbers.

The coefficients for the static linear model in Section 7.3 are calculated using a Bayesian network model. Each cluster contains different coefficients for the A, B, C and D matrices.

Name	Abbr.	Index (b)	Cluster (Cl(b))
Bolton	BOL	1	4
Bury	BUR	2	3
Manchester	MAN	3	4
Oldham	OLD	4	4
Rochdale	ROC	5	4
Salford	SAL	6	4
Stockport	STO	7	3
Tameside	TAM	8	4
Trafford	TRA	9	2
Wigan	WIG	10	3

Table 7.1: The LADs in Greater Manchester, along with their abbreviations, the index assigned to them, and their corresponding cluster.

Figure 7.2: The local authority clusters in Greater Manchester.

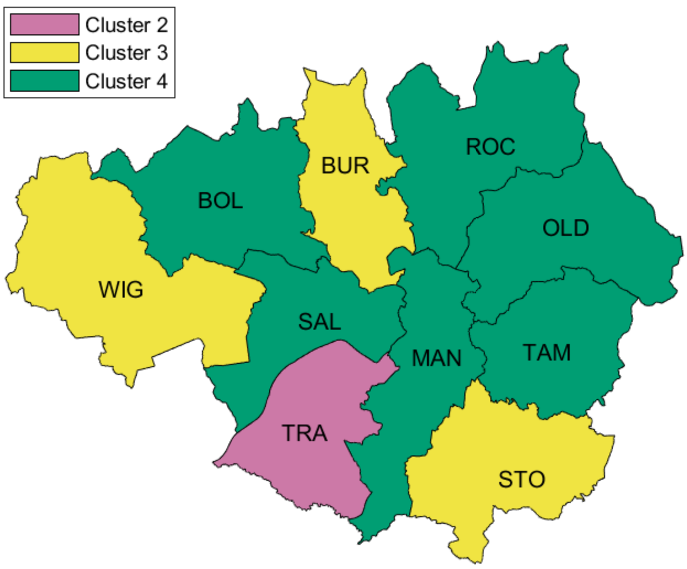
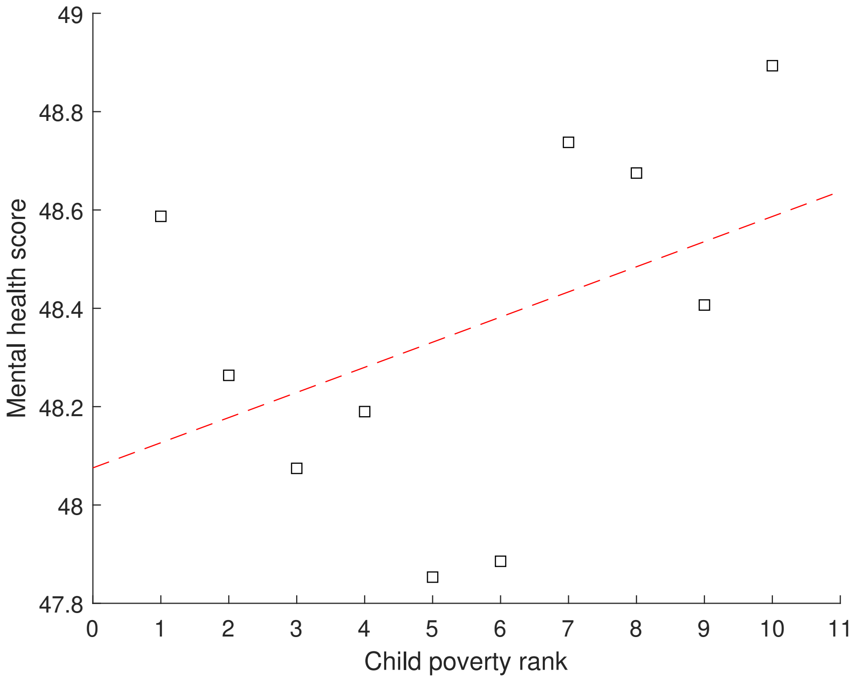


Figure 7.3: How the slope index of inequality is calculated. The black points are example data and the red dashed line is the regression line.



7.2.3 Slope index of inequality

The slope index of inequality (abbreviated SII) indicates the difference in SF-12 mental health scores (MCS) between the most impoverished areas and the least impoverished areas. In the case of the GMCA problem, the metric used is child poverty and the areas being ranked are LAs.

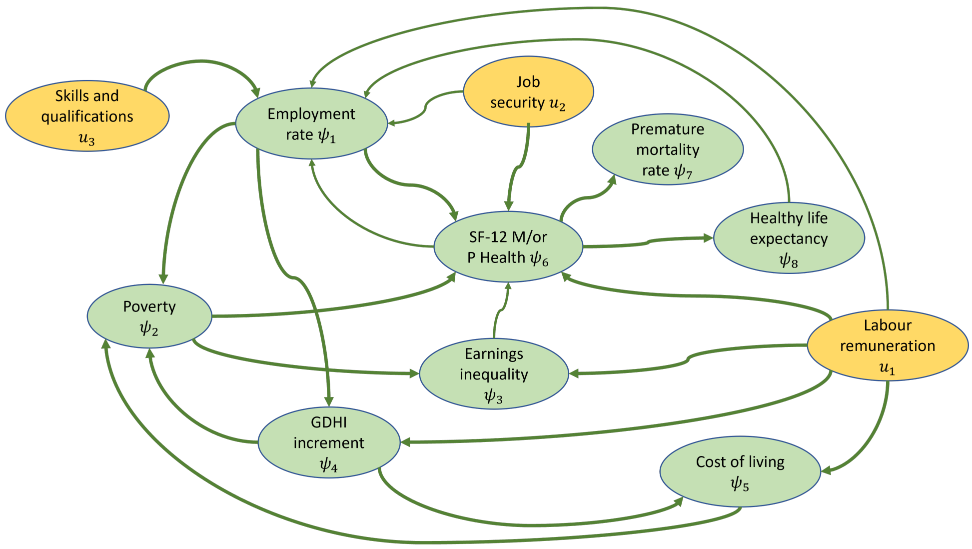
The LAs are ranked from highest child poverty to lowest child poverty. A linear regression between the ranked LAs and their corresponding MCS is performed. The difference in MCS between the regression model at the highest ranked LA (i.e. ten) and the lowest ranked LA (one) is the slope index of inequality. Figure 7.3 shows the slope index regression line plotted over some example data for the ten LAs in GM.

7.3 System model

The system model comprises two main concepts: firstly, the systems map and associated definitions, which can be found in [166], and the system equations, which were developed in [167], using the aforementioned systems map. This section will describe the work done by these other researchers as a background for the rest of the work in this chapter.

7.3.1 Systems map

Figure 7.4: The systems map as it appears in the WS4 technical document.



The original systems map is shown in Figure 7.4. This shows causal relationships between economic and health indicators. The yellow input nodes are expressed in the systems map as u_i where i is an index. The green output (indicator) nodes are expressed as ψ_i. These are found in [166] and are copied here for clarity.

1. *Employment rate:* The percentage employment rate of people aged between 16 and 64.

2. *Child poverty:* Percentage of children living in low-income households.

3. *Cost of living:* Percentage of households that are fuel-poor in an LA.

4. *Gross domestic household income:* Gross disposable household income per

head.

5. *Earnings inequality:* Ratio of weekly earnings between 20th and 80th percentile.

6. *Mental health:* SF12 mental health values.

7. *Premature mortality rate:* Directly age-standardised mortality rate for all deaths registered in the respective calendar years, in people aged under 75. Directly standardised rate per 100 000.

8. *Healthy life expectancy:* Healthy life expectancy at birth.

The output nodes are shown in Table 7.2, with their three-letter abbreviations, their assigned index, and the subsystem they belong to.

Name	Abbr.	Index	Subsystem	Typical range
Employment rate	EPR	1	1	58.6% - 89.2%
Child poverty	PVT	2	2	10.64% - 32.5%
Cost of living	FPR	3	2	16.9% - 64.8%
Gross domestic household income	DHI	4	2	£9501 - £24 445
Earnings inequality	EIQ	5	2	1.82 - 2.57
Mental health	MCS	6	2	43.2 - 54.2
Premature mortality rate	PMR	7	2	274 - 780
Healthy life expectancy	HLE	8	2	49.29 years - 71.2 years

Table 7.2: Output nodes from the systems map.

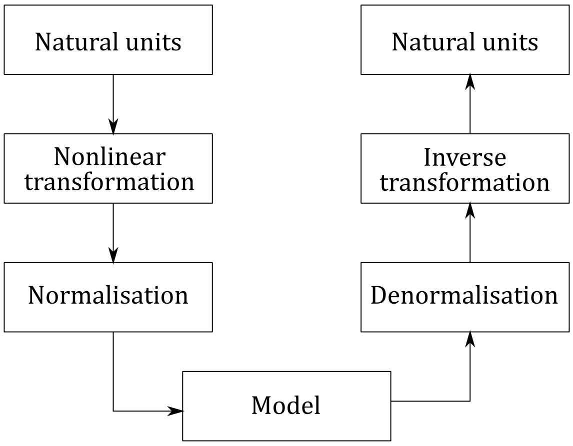
Name	Abbr.	Index	Subsystem
Labour remuneration	PLW	1	2
Job security	JPC	2	2
Skills and qualifications	SAQ	3	1

Table 7.3: Input nodes from the systems map.

7.3.2 System equations

Figure 7.5 shows the process for inputting and outputting data to and from the model. Natural units (for example percentage of people with NVQ2+ skills or percentage of people on below the real living wage) are used as an input. These undergo a nonlinear transformation shown in Section 7.3.3. They are then normalised, following which they can be entered into the model. The model returns a corresponding

Figure 7.5: The process used for transforming and normalising data for use in the GMCA model.



output vector, which is then denormalised. The denormalised values undergo the inverse transformation to the one used for model input, and the natural units are returned. These can be used in the analysis of the model or optimization results.

7.3.3 Transformations, normalisation and denormalisation

The values going into and coming out of the linear model need to be normalised or denormalised before they can be passed through the transformations.

The normalised values are calculated using

$$\phi_{norm} = \frac{\phi_{trans} - \mu}{\sigma} \tag{7.1}$$

and the denormalised values are calculated using

$$\phi_{denorm} = \sigma\phi_{model} + \mu \tag{7.2}$$

The σ and μ values are shown in Table 7.4. The nonlinear and inverse transformations described in Section 7.3.3 are also shown here.

Variable	μ	σ	Non-linear transformation	Inverse transformation
u_1	-1.1470	0.3060	$log(\frac{\phi_{raw}}{100-\phi_{raw}})$	$\frac{100e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
u_2	-3.0625	0.4554	$log(\frac{\phi_{raw}}{100-\phi_{raw}})$	$\frac{100e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
u_3	0.9039	0.4370	$log(\frac{\phi_{raw}}{100-\phi_{raw}})$	$\frac{100e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
ψ_1	1.0756	0.2981	$log(\frac{\phi_{raw}}{100-\phi_{raw}})$	$\frac{100e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
ψ_2	-0.9247	0.2773	$log(\frac{\phi_{raw}}{100-\phi_{raw}})$	$\frac{100e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
ψ_3	-2.1000	0.1388	$log(\frac{\phi_{raw}}{100-\phi_{raw}})$	$\frac{100e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
ψ_4	9.7798	0.2701	$log(\phi_{raw})$	$e^{\phi_{trans}}$
ψ_5	0.2081	0.1388	$log(\phi_{raw} - 1)$	$1 + e^{\phi_{trans}}$
ψ_6	1.2153	0.2057	$log(\frac{\phi_{raw}-12}{60-\phi_{raw}})$	$12 + \frac{48e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
ψ_7	-5.6569	0.2064	$log(\frac{\phi_{raw}}{100\,000-\phi_{raw}})$	$\frac{100\,000e^{\phi_{trans}}}{1+e^{\phi_{trans}}}$
ψ_8	4.1368	0.0538	$log(\phi_{raw})$	$e^{\phi_{trans}}$

Table 7.4: The μ and σ values for normalising and denormalising values in the GMCA model.

7.3.4 Static model

The GMCA model is a static model and takes the form

$$\boldsymbol{\psi} = A\boldsymbol{\psi} + B\mathbf{u} + C. \tag{7.3}$$

Equation 7.4, from [167] shows the equation for the static model. See Appendix C for the coefficients for this model.

$$
\begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \\ \psi_6 \\ \psi_7 \\ \psi_8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{2,1} & 0 & A_{2,3} & A_{2,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{5,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{6,1} & A_{6,2} & 0 & 0 & A_{6,5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{7,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{8,6} & 0 & 0 \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \psi_5 \\ \psi_6 \\ \psi_7 \\ \psi_8 \end{bmatrix} + \begin{bmatrix} B_{1,1} & B_{1,2} & B_{1,3} \\ 0 & 0 & 0 \\ B_{3,1} & 0 & 0 \\ B_{4,1} & B_{4,2} & 0 \\ B_{5,1} & 0 & 0 \\ B_{6,1} & B_{6,2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \end{bmatrix} \tag{7.4}
$$

7.4 Optimization problem

There are three objectives in the optimization problem:

1. f_1 Maximise the population weighted mean healthy life expectancy.
2. f_2 Minimise the mental health inequality – this is measured as the slope index of inequality measured by child poverty rates between the LAs with the highest MCS and those with the lowest MCS.
3. f_3 Minimise the proportion of the budget spent.

And there are seven inequality constraints:

- g_1: The total amount of money spent must be equal to or lower than the total budget G.
- g_2: The inputs to subsystem one must be equal to global variable z_1. This constraint is relaxed to allow the optimizer to converge.

- g_3: The inputs to subsystem two must be equal to global variable z_2. This constraint is relaxed to allow the optimizer to converge.

- g_4, g_5: Solutions must be excluded where the individual subsystems exceeded the amount of money provided by the system-level optimizer. In other words, the subsystems are permitted to use up less money than what is provided by the optimizer, but not exceed it.

- g_6, g_7: Additionally, some other inequality constraints were added to help avoid issues with the model at extreme ends of PLW and SAQ.

$\mathbf{x}_{a,b}$ is the investment in subsystem a, local authority b, and the global variable $\mathbf{z}_a$ is the investment in subsystem a. The variables $\mathbf{x}$ and $\mathbf{z}$ are bound constrained by the total budget allocated to the problem, G. Additionally, the variables must be positive (in other words, areas may only be invested in, not divested).

This problem will be solved using the MDF architecture.

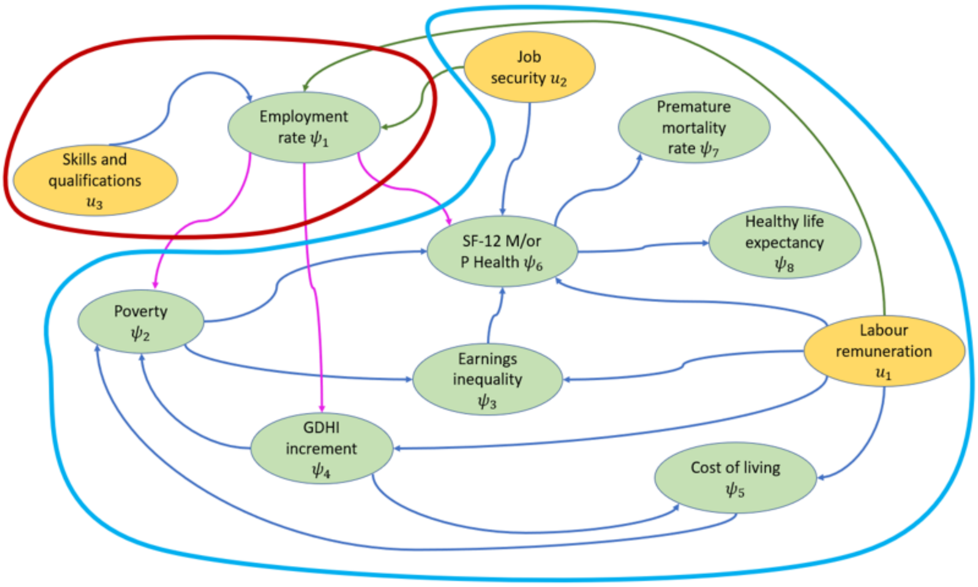
7.4.1 The MDF problem

The full MDF optimization problem is expressed as follows.

$$
\begin{aligned}
\max \quad & \frac{\sum_{b=1}^{10} \lambda_b y_{2,b,2}}{\sum_{b=1}^{10} \lambda_b} \\
\min \quad & SII(y_{2,1,2}, ..., y_{2,10,2}) \\
\min \quad & z_1 + z_2 \\
\text{w.r.t} \quad & \mathbf{x}, \mathbf{z} \\
\text{s.t.} \quad & g_1 \equiv z_1 + z_2 \leq G \\
& g_2 \equiv |\sum_{i=1}^{10} x_{1,i} - z_1| \leq 1 \\
& g_3 \equiv |\sum_{i=1}^{10} x_{3,i} - z_2| \leq 1 \\
& g_4 \equiv \sum_{i=1}^{10} x_{i,1} \leq z_1 \\
& g_5 \equiv \sum_{i=1}^{10} x_{i,2} \leq z_2 \\
& g_6 \equiv \max_{1 \leq i \leq 10, j=1,2} (u_{i,j}(\mathbf{x}, \mathbf{z})) \leq 99 \\
& g_7 \equiv \min_{1 \leq i \leq 10, j=1,2} (u_{i,j}(\mathbf{x}, \mathbf{z})) \geq 1 \\
& 0 \leq \mathbf{z} \leq G \\
& 0 \leq \mathbf{x} \leq G \\
\text{where} \quad & \mathbf{y}_1 = model_1(\mathbf{x}_1, \mathbf{y}_2) \\
& \mathbf{y}_2 = model_2(\mathbf{x}_2, \mathbf{y}_1)
\end{aligned}
\tag{7.5}
$$

where λ_b is the population of local authority b and SII is the slope index of inequality calculation. The linking variables are expressed as $y_{a,b,c}$ where c is the linking variable number (for example, the second linking variable in subsystem two for LA eight would be $y_{2,8,2}$).

Figure 7.6: The systems map with subsystem partitions (outlined in red and blue for subsystems one and two respectively).



7.4.2 Systems map partitioning

For the purposes of the GMCA MO-MDO problem, the systems map is split into two subsystems, shown in Figure 7.6. The nodes enclosed in a red boundary are nodes in subsystem one (called adult skills) and the nodes enclosed in the blue boundary are subsystem two (called local growth and place).

Additionally, two connections had to be removed from the systems map for the purposes of model fitting. These are the HLE to EPR connection; and the MCS to EPR connection. Removing these connections turned the systems map from a cyclic to an acyclic map, and permitted the fitting of a Bayesian network model, from [167].

To accommodate this partition, the systems map equations 7.3 and 7.4 are reformulated to include the linking variables from the other subsystems. This is shown in Equation 7.6.

$$\boldsymbol{\psi} = A\boldsymbol{\psi} + B\mathbf{u} + C + D\mathbf{y} \tag{7.6}$$

$\boldsymbol{\psi}_i$ is the output from subsystem i. In some cases, a $\boldsymbol{\psi}_i$ may be either a linking variable (see for example Equation 7.9) or just a variable used in the objective or constraint functions in the optimization problem.

Subsystem 1 is expressed as

$$\boldsymbol{\psi}_1 = y_1 = B_{3,1}u_3 + D_{1,1}y_{2,1} + D_{1,2}y_{2,2} \tag{7.7}$$

Subsystem 2 is expressed as

$$\begin{bmatrix}\psi_2\\\psi_3\\\psi_4\\\psi_5\\\psi_6\\\psi_7\\\psi_8\end{bmatrix} = \begin{bmatrix}0 & A_{2,3} & A_{2,4} & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0 & 0\\A_{5,2} & 0 & 0 & 0 & 0 & 0 & 0\\A_{6,2} & 0 & 0 & A_{6,5} & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & A_{7,6} & 0\\0 & 0 & 0 & 0 & 0 & A_{8,6} & 0\end{bmatrix}\begin{bmatrix}\psi_2\\\psi_3\\\psi_4\\\psi_5\\\psi_6\\\psi_7\\\psi_8\end{bmatrix} + \begin{bmatrix}0 & 0\\B_{3,1} & 0\\B_{4,1} & B_{4,2}\\B_{5,1} & 0\\B_{6,1} & B_{6,2}\\0 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}u_1\\u_2\end{bmatrix} + \begin{bmatrix}C_2\\C_3\\C_4\\C_5\\C_6\\C_7\\C_8\end{bmatrix} + \begin{bmatrix}D_2\\0\\0\\0\\D_6\\0\\0\end{bmatrix} y_1 \tag{7.8}$$

Additionally,

$$\begin{aligned}y_{2,1} &= u_1\\y_{2,2} &= u_2\end{aligned} \tag{7.9}$$

7.4.3 Subsystem analyses

The subsystem analyses are reliant on the static linear model described in Section 7.3. This model is partitioned as shown in Figure 7.6. This model becomes a function of the local and linking variables, expressed as $model(\bullet)$. Additionally, the input nodes u are a multiplication of the local variables x and an effectiveness coefficient β, which is derived from real-world data.

Subsystem one, adult skills, is written

$$\begin{aligned} u_{3,b} &= \beta_1 x_{1,b} \\ s_{1,b} &= model_1\left(x_{1,b}, \mathbf{y}_{2,b}\right) \\ \mathbf{y}_{1,b} &= s_{1,b}. \end{aligned} \tag{7.10}$$

Subsystem two, local growth and place, is written

$$\begin{aligned} u_{1,b} &= \beta_2 x_{2,b} \\ u_{2,b} &= \beta_3 x_{3,b} \\ [s_{6,b}, s_{7,b}] &= model_2\left(x_{2,b}, x_{3,b}, \mathbf{y}_{1,b}\right) \\ \mathbf{y}_{2,b} &= [s_{6,b}, s_{7,b}, u_{1,b}, u_{2,b}]. \end{aligned} \tag{7.11}$$

The full XDSM for the problem is shown in Figure 7.7.

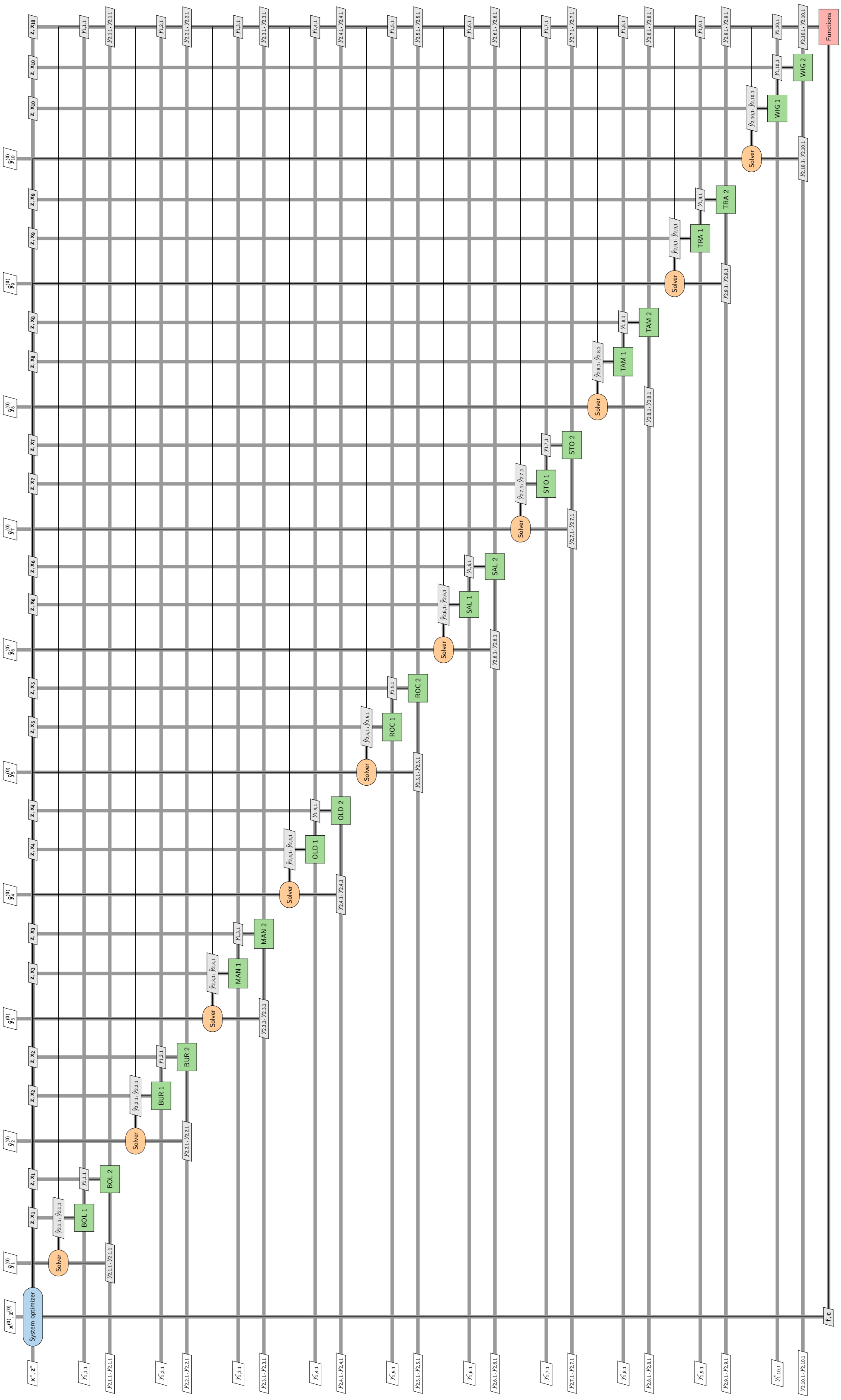


Figure 7.7: The full XDSM for the GMCA optimization problem using the MDF architecture.

7.5 Preliminary analysis and decision modelling

7.5.1 Preliminary analysis

Figures 7.8, 7.9 and 7.10 show the preliminary analyses for clusters two, three and four respectively. In these plots, the values for skills and qualifications (labelled SAQ) and labour remuneration (labelled PLW) are varied from the baseline to a 'better' value (this is upwards for SAQ and downwards for PLW). Each of the subplots shows the values from the output nodes for each cluster. The units for these can be found in the last column of Table 7.2.

These preliminary analyses are important because they show clearly how the model is expected to work and what it is expected to output given a certain value or change in SAQ and PLW. This is useful for 'sanity checking' the optimization results.

7.5.2 Determining β and γ

The input values u_i are calculated using the equation

$$u_{i,j} = \gamma_{i,j} + \beta_{i,j} x_{i,j} \tag{7.12}$$

where $\gamma_{i,j}$ is the baseline level for local authority i in subsystem j and $\beta_{i,j}$ is the effectiveness of funding placed into local authority i for subsystem j. γ and β are calculated using real-world data obtained from Greater Manchester Combined Authority. In addition to the β and γ values, the u output is saturated between 0 and 100 for all u values. A diagram of the structure of the problem showing how the u, x and z variables are related is given in Figure 7.11.

Figure 7.12 shows how u can be interpolated or extrapolated.

Figure 7.8: A preliminary analysis of the different output nodes in cluster two.

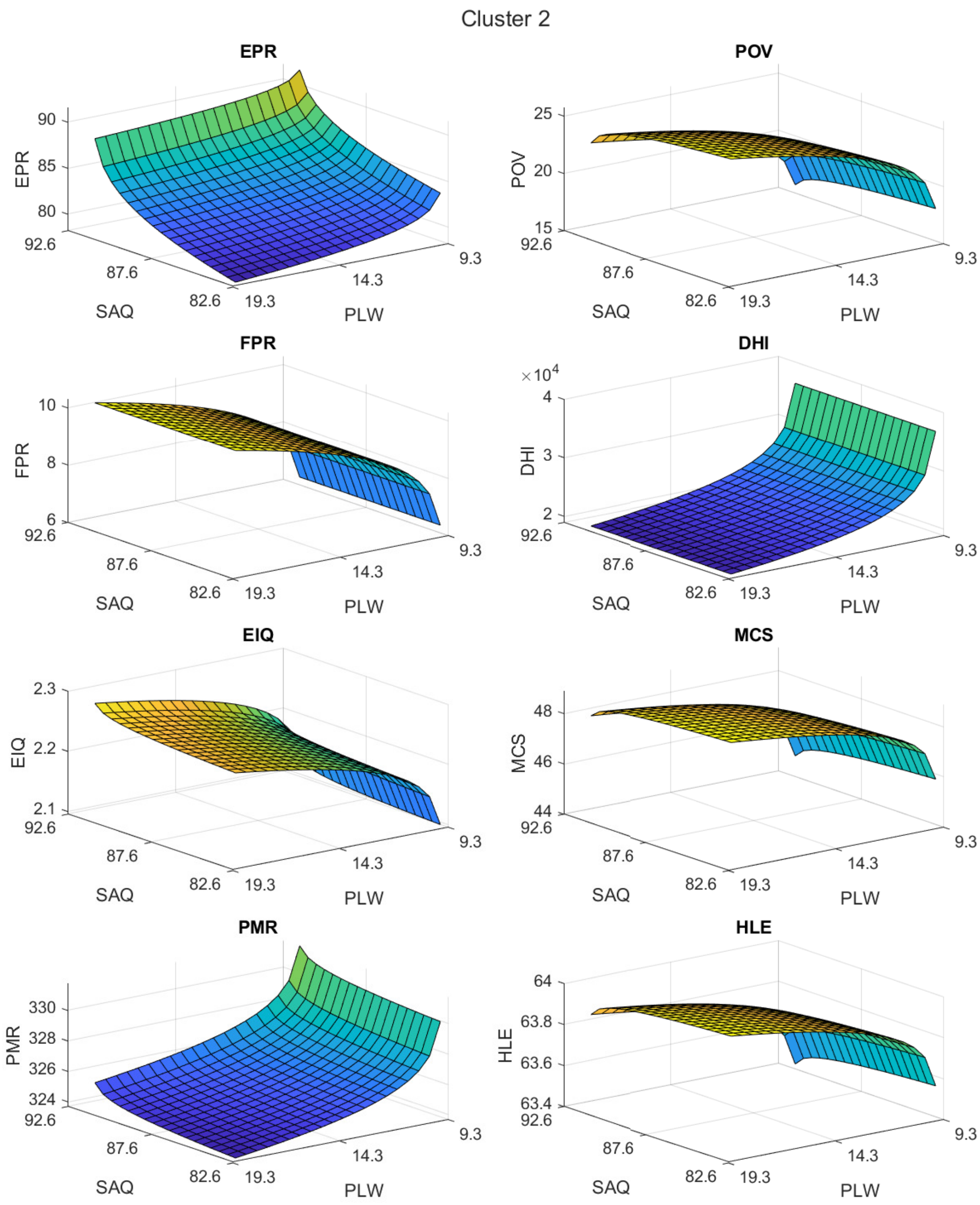


Figure 7.9: A preliminary analysis of the different output nodes in cluster three.

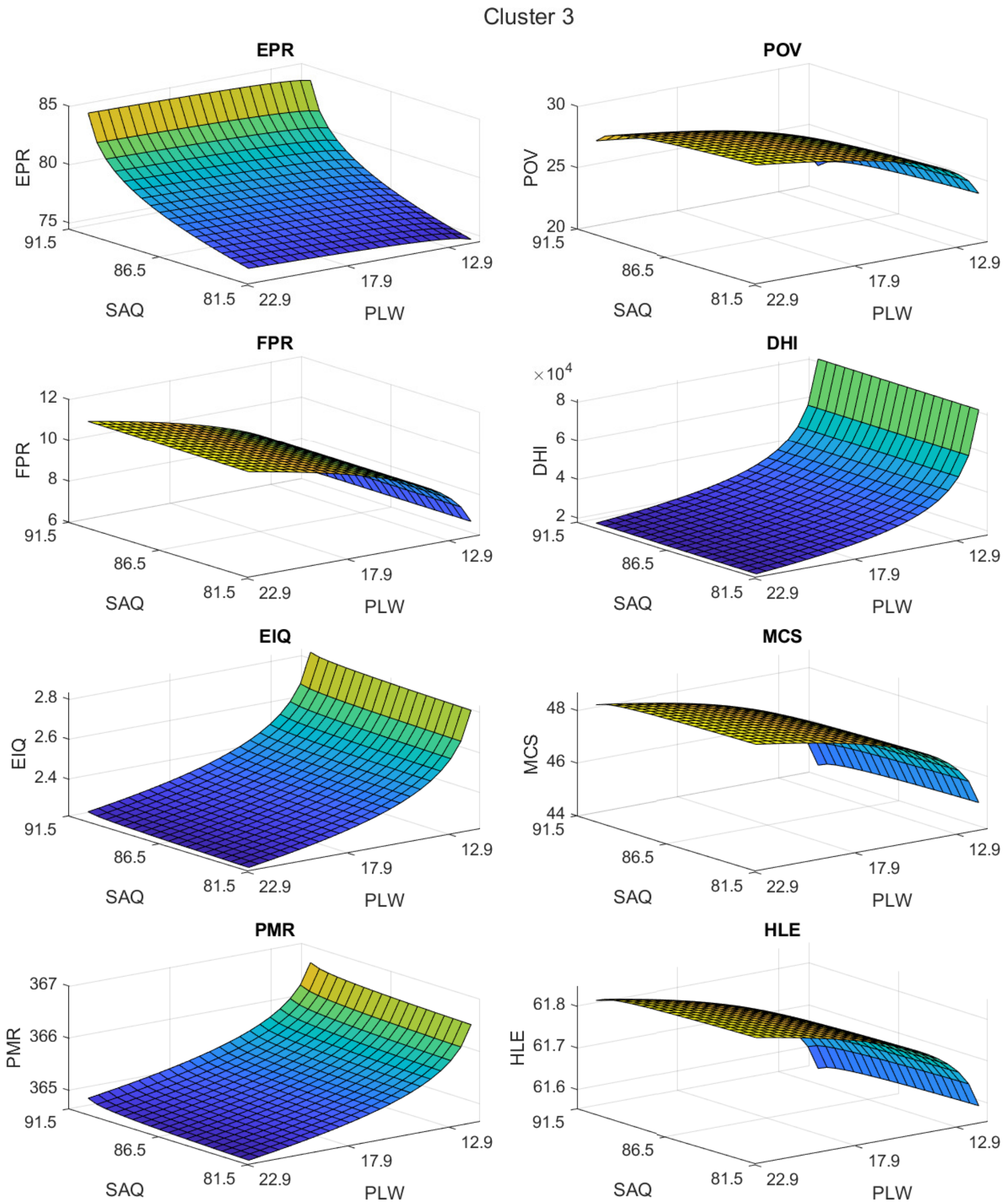


Figure 7.10: A preliminary analysis of the different output nodes in cluster four.

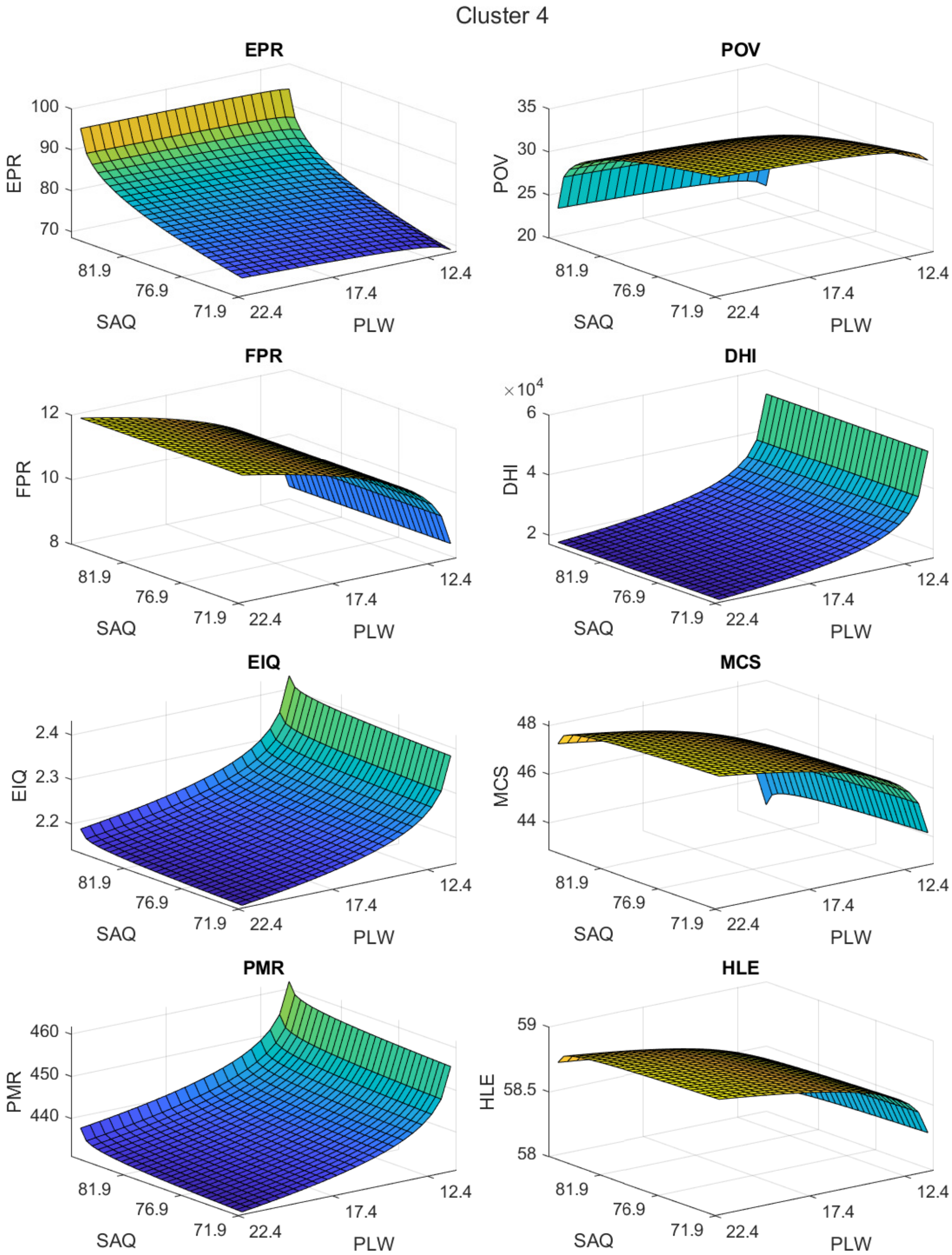


Figure 7.11: A simplified structure of the problem (one LA) showing how the u, x and z variables are related.

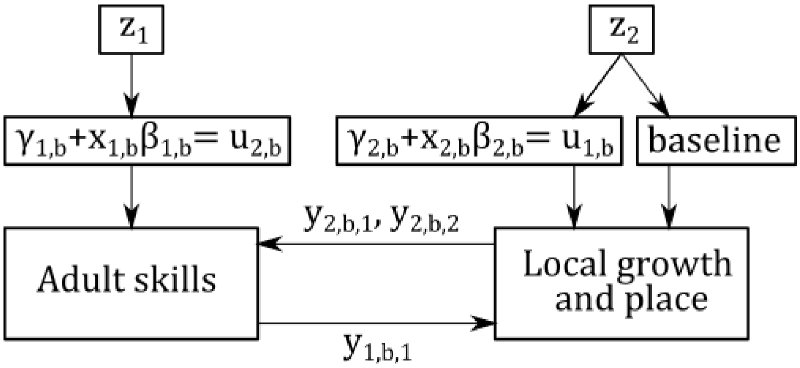
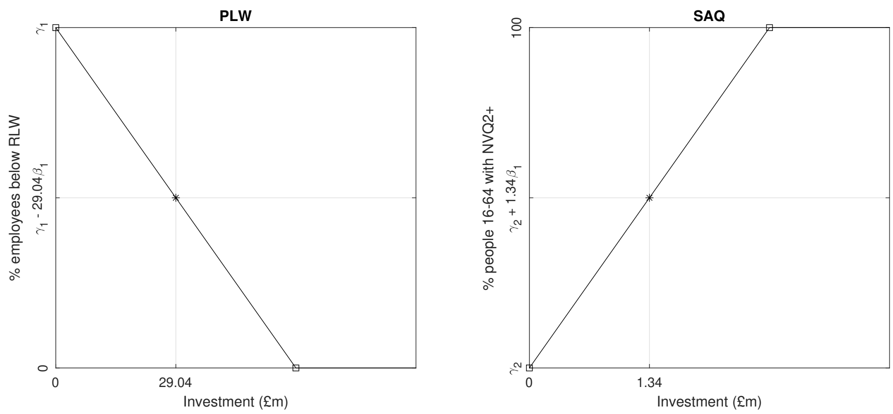


Figure 7.12: An example of how u_1 and u_3 are calculated using β and γ. Note the saturation in both plots.



Adult skills

Two tables were provided to us by GMCA relating to skills and qualifications. Firstly, the amount of funding provided for each adult skills level, excluding everything below NVQ (national vocational qualification) levels two and three, for academic years beginning 2020, 2021 and 2022. This is shown in Table 7.5.

	Level 2	Level 3	Total
2020/21	[redacted]	[redacted]	[redacted]
2021/22	[redacted]	[redacted]	[redacted]
2022/23	[redacted]	[redacted]	[redacted]

Table 7.5: [TABLE REDACTED] Funding provided for NVQ levels two and three in Greater Manchester.

Table 7.6 shows the number of achievements (qualifications completed successfully) for levels two and three. In other words, this is the number of people who increased their qualification level.

	Level 2	Level 3	Total
2020/21	[redacted]	[redacted]	[redacted]
2021/22	[redacted]	[redacted]	[redacted]
2022/23	[redacted]	[redacted]	[redacted]

Table 7.6: [TABLE REDACTED] Number of achievements for NVQ levels two and three in Greater Manchester.

The β_1 for the skills and qualifications input node, u_3, indicates the effectiveness of the funding provided to the subsystem. When multiplied with the funding provided to the subsystem x_1, it provides the percentage by which the number of people in the LA with NVQ levels two and up increases.

Table 7.8 shows the number of people in each LAD with at least NVQ two and above. The data for number of people aged 16–64 in each LA is obtained from [16] for the year 2022.

LAD	β_1	γ_1
BOL	0.03671	75.6
BUR	0.04035	83.1
MAN	0.03481	71.7
OLD	0.03423	70.5
ROC	0.03535	72.8
SAL	0.03807	78.4
STO	0.03943	81.2
TAM	0.03593	74.0
TRA	0.04069	83.8
WIG	0.03817	78.6

Table 7.7: β_1 and γ_1 values for skills and qualifications funding in GMCA.

LAD	Number of people aged 16–64	Percentage of people with NVQ2+	Estimated number of people aged 16–64 with NVQ2+*	% of people aged 16–64 with NVQ2+*	Proportional number of people 16–64 with NVQ2+*	New % people 16–64 with NVQ2+*
BOL	183 000	75.6%	138 348	9.75	2041	76.7
BUR	119 900	83.1%	99 637	7.02	1470	84.3
MAN	400 900	71.7%	287 445	20.26	4240	72.8
OLD	149 600	70.5%	105 468	7.43	1556	71.5
ROC	139 900	72.8%	101 847	7.18	1502	73.9
SAL	187 500	78.4%	147 000	10.36	2169	79.6
STO	180 300	81.2%	146 404	10.32	2160	82.4
TAM	145 600	74.0%	107 744	7.59	1589	75.1
TRA	145 200	83.8%	121 678	8.58	1795	85.0
WIG	207 700	78.6%	163 252	11.51	2408	79.8

Table 7.8: Number of people in each LAD with education level NVQ two and above for the different LAs in GM. Table headings with a * indicate calculated values, headings without a * are obtained from the Office of National Statistics [16]

LAD	Total jobs*	Jobs below RLW	% jobs below RLW	% below RLW jobs in LA*	Total people uplifted in LA*	New % jobs below RLW*
BOL	94203	13000	13.8	9.09	2340	11.3
BUR	64220	7000	10.9	4.90	1260	8.94
MAN	371134	36000	9.7	25.17	6480	7.95
OLD	77348	14000	18.1	9.79	2520	14.8
ROC	67873	15000	22.1	10.49	2700	18.1
SAL	123596	11000	8.9	7.69	1980	7.30
STO	98361	12000	12.2	8.39	2160	10.0
TAM	51546	10000	19.4	6.99	1800	15.9
TRA	110092	12000	10.9	8.39	2160	8.94
WIG	91549	13000	14.2	9.09	2340	11.6

Table 7.9: Number of jobs below the RLW in different LAs in GM. Table headings with a * indicate calculated values, headings without a * are obtained from the Office of National Statistics [17]

Labour remuneration

Labour remuneration is the number of people who have had their wages increased from the National Living Wage (NLW) to the Real Living Wage (RLW). The NLW is the current hourly wage that must be paid to all workers age 23 and over – at the time of writing (March 2024), this is £10.42 per hour for workers outside of London. The RLW is a calculation made based on a basket of goods and reflects the financial reality of living in the United Kingdom. Outside of London, this is £10.90 per hour.

GMCA has introduced the Greater Manchester Good Employment Charter, which is a scheme that employers sign up to, that ensures all employees in an accredited company are guaranteed a certain standard of work environment, including secure work, livable pay, fair recruitment practices, and a workplace that encourages good physical and mental health. When a company or organisation joins the Good Employment Charter, it is required that all employees in the organisation are paid at least the RLW. This means that employees' salary is uplifted.

GMCA spent [redacted] from the launch of the Good Employment Charter (January 2020) until June 2023. In this time, the number of employees who had their wage uplifted from the NLW to the RLW was [redacted].

LAD	β_2	γ_2
BOL	-1.6384	22.4
BUR	-1.6750	22.9
MAN	-1.0679	14.6
OLD	-1.6896	23.1
ROC	-1.4190	19.4
SAL	-1.1923	16.3
STO	-1.3020	17.8
TAM	-2.0846	28.5
TRA	-1.4117	19.3
WIG	-1.7189	23.5

Table 7.10: β_2 and γ_2 values for labour remuneration funding in GMCA.

Job security

Unlike the previous two β values, GMCA did not have any data linking investment to outcomes for increasing job security. This was resolved by setting all u values in the Job Security node to the baseline for each LA. In practical terms, this was the percentage of employed people who were in non-permanent work. These baselines are shown in Table 7.11.

LAD	Baseline job security (%)
BOL	3.3
BUR	4.8
MAN	6.4
OLD	3.9
ROC	3.7
SAL	4.5
STO	6.1
TAM	5.6
TRA	5.5
WIG	4.0

Table 7.11: The baseline percentage of employed people in each LA in non-permanent employment.

7.5.3 Determining G

To determine G, the following pieces of information were obtained:

- The amount of money spent on skills and qualifications by GMCA
- The amount of money spent on labour remuneration
- The amount of money spent on job security

G was determined by summing the total amount of funding allocated to the input values x_1, x_2 and x_3.

$$G = G_1 + G_2 + G_3 \tag{7.13}$$

The data used for calculating the βs in Section 7.5.2 is also used for calculating G.

The amount of funding provided to input node u_3 adult skills is calculated as the average funding for the years 2020/21 and 2021/22.

G_1	[redacted]
G_2	[redacted]
G_3	[redacted]
Total	[redacted]

Table 7.12: [TABLE REDACTED] The amount of funding provided by GMCA for the three input nodes.

7.6 Experimental setup

The optimization was executed using Python. The optimizer used was NSGA-II, as implemented in PyMoo [158]. The number of generations in NSGA-II were set to 500, and the population was 100. The crowding degree of mutation (η_m) and crossover (η_c) were 20 and 15 respectively. The crossover probability was 15% and the mutation probability was 4.55% (i.e. $\frac{1}{22}$). The solver used for the disciplinary analyses was the Newton solver, as implemented in SciPy [159].

The optimization was run 21 times and the solution with the median hypervolume was selected for analysis in this chapter. The reference point for the hypervolume was $\{-48.5, 0.5, 20\}$.

7.7 Results

7.7.1 Baselines

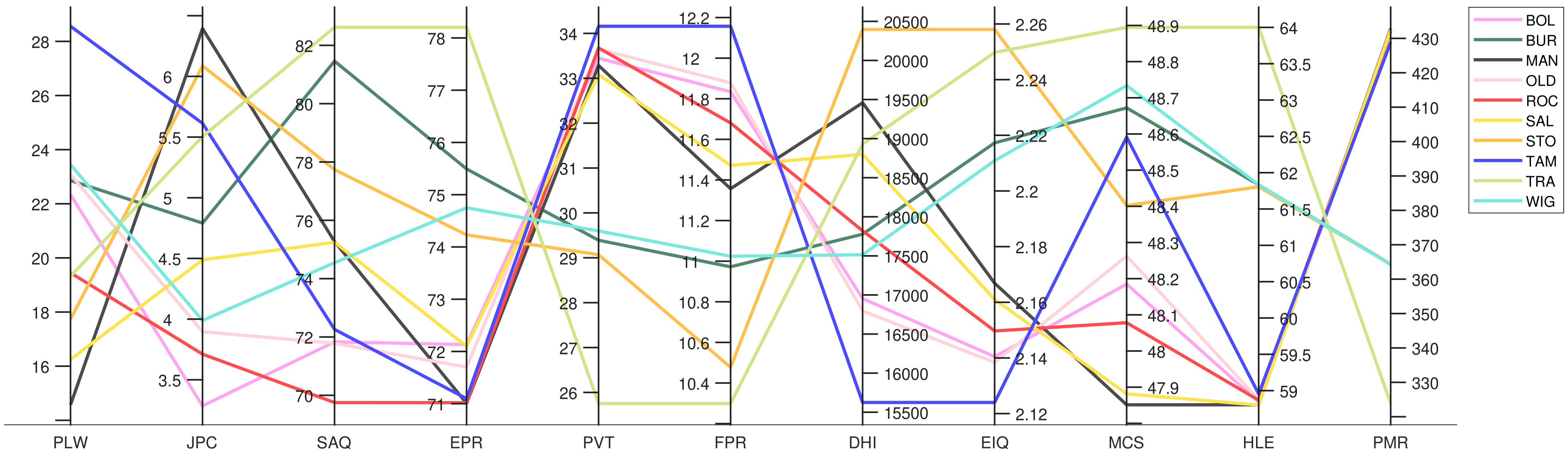
In this subsection, the baselines are presented for reference for the optimization results subsection 7.7.2. Baselines are important because they demonstrate a system's outputs given either a known intervention (such as continuing investment into a combined local authority at the same quantity as previously), or no intervention. This permits the comparison of results against other results, as well as the baseline. In this case, the baseline is calculated by setting the system inputs at the existing, measured values.

The baseline values for the input and output nodes are shown in Figure 7.14. It can be seen that Trafford performs the best in many metrics, with the highest values for skills and qualifications (i.e. more people in Trafford are educated to NVQ2+ than in any other LA in GMCA), has the highest employment rate, lowest child and fuel poverty, highest average SF-12 mental health score and healthy life expectancy, and lowest premature mortality rate. However, it has one of the highest income inequality scores, behind Stockport. This is expected because, as shown in Figure 7.2, Trafford is the only Cluster 2 LA. Correspondingly, it can be seen that the Cluster 4 LAs perform badly in metrics such as premature mortality rate (all Cluster 4 LAs have a PMR of around 430), healthy life expectancy, and mental health scores. One notable example to this is Tameside's mental health scores – Tameside (Cluster 4) performs badly in most metrics, but has a higher SF-12 mental health score than Stockport, which is a Cluster 3 LA. These baselines can also be seen in the choropleth in Figure 7.13.

One thing should be noted in the baseline scores. Firstly, the healthy life expectancies are clustered around three points, corresponding to the three clusters. This is about 59 for Cluster 4, 61.8 for Cluster 3, and 64 for Cluster 2. This happens because in the systems map (see Figure 7.4) the healthy life expectancy node is influenced only *through* another input node – mental health. Because of the way the static models are set up, this means that the input signals (through SAQ and PLW) are multiplied twice by a small value (< 1). As a result, the HLE node does not deviate much from the 'centre point' for the cluster. This is also the case for premature mortality rates. This effect is also seen in the optimization results.

Figure 7.13: The choropleth showing the baseline node values for each of the LAs in GMCA.

Figure 7.14: The baseline values for the GMCA system. Here, the baseline is when no additional investment is entered into the system (for example in Figure 7.12 where the investment is £0).



7.7.2 Optimization results

In this section the results of the optimization process are presented. Three key results are introduced and the differences between them are discussed. These three key results are:

- The solution that produces the lowest health inequality (r_1)
- The solution that produces the highest weighted average mental health across Greater Manchester (r_2)
- The lowest cost solution produced by the optimizer (r_3).

These results are summarised in Table 7.13. The 'colour' column indicates what colour these points are on the corresponding plots – the baseline is not shown, so it has not been allocated a colour. These results are also highlighted in Figure 7.15, where the solutions are shown in the objective space. The black points are non-key, nondominated solutions.

Solution	Colour	Cost (£m)	Average MCS	SII
r_1	Red	15.4003	48.6008	2.5088×10^{-7}
r_2	Yellow	2.5219	48.6522	0.3040
r_3	Blue	2.0823	48.6506	0.2936
Baseline		0	48.3017	0.8853

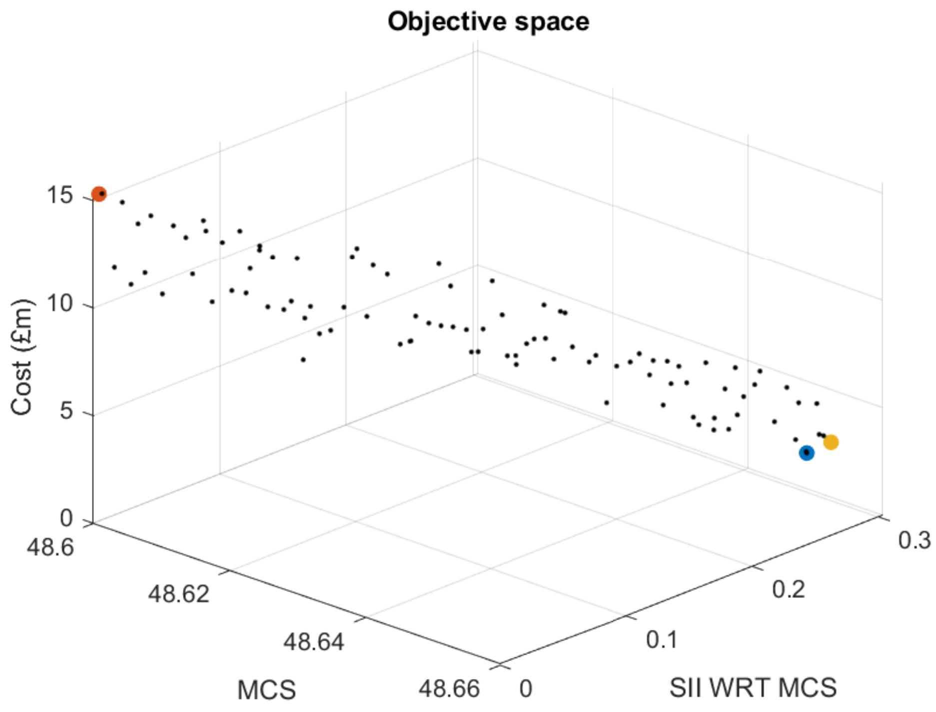
Table 7.13: The three key solutions and the baseline value.

Design variables

For the application of the multi-objective MDF architecture to the GMCA problem, it is important to present the design variables to best illustrate the cost-effectiveness of each solution, as well as the differences between them. Note here that the design variable units are in millions of pounds (£m).

A breakdown by LA of the funding allocations for the three key solutions is shown using a parallel coordinates plot in Figure 7.17. There are a few things to note here. Firstly, the cheapest solution and the solution with the highest average mental health are very similar; this negative relationship between cost and mental health is also apparent in the matrix scatter plot, Figure 7.16. Secondly, the skills

Figure 7.15: The objective space following the optimization process. The red, yellow and blue points are notable solutions. The red point represents the solution with the lowest inequality, the yellow point represents the solution that produces the highest weighted average mental health, and the blue solution is the lowest cost solution.



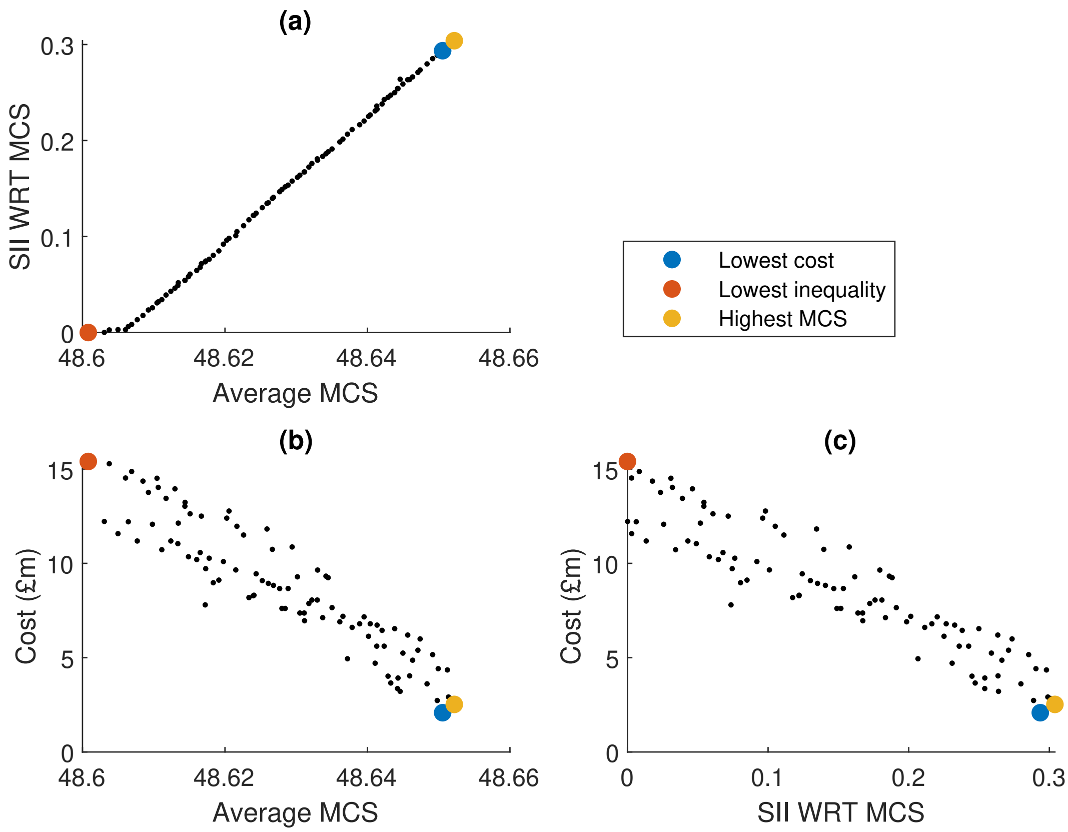
and qualifications subsystem was allocated more overall funding than the labour remuneration subsystem in all three key solutions. Finally, Trafford received a large increase in funding in the labour remuneration subsystem, while Manchester, Salford and Stockport received the most funding in the skills and qualifications subsystem.

The convergence of design variables for the three key solutions are shown in Figure 7.18. It can be seen that the design variable values are highest in the first 20 iterations or so, and then drop to below £5m for the rest of the iterations.

Figures 7.19 and 7.20 show the funding distribution between the two subsystems (summed over all LAs). In both figures, the horizontal axis shows the funding going to the 'Skills and qualifications' discipline, while the vertical axis shows the funding going to the 'Local growth and place' discipline. Both of these are in millions of pounds (£m).

In Figure 7.19, a larger and more saturated point on the scatter plot indicates

Figure 7.16: A matrix scatter plot showing the behaviour of the objective solutions following optimization. The top left plot shows the average mental health score plotted against the slope index with respect to mental health (a). The bottom right shows average mental health against cost (b). The bottom left shows slope index against cost (c).



a solution that produces a larger average mental health score over the system. It can be seen that the best solutions are concentrated in the lower left quadrant of the plot – in other words, these are the solutions that have very little funding. It can also be seen that the lowest cost solution and the highest average mental health solutions are close together.

In Figure 7.20, a larger and more saturated point indicates a solution where the inequality of mental health is lower. In contrast to Figure 7.19, the best solutions by inequality are concentrated in the top right quadrant of the plot. The solution with the lowest inequality by SII is marked with a red point.

Figure 7.17: The parallel coordinates plot for funding for skills and qualifications (top) and labour remuneration (bottom). Only the key solutions are shown.

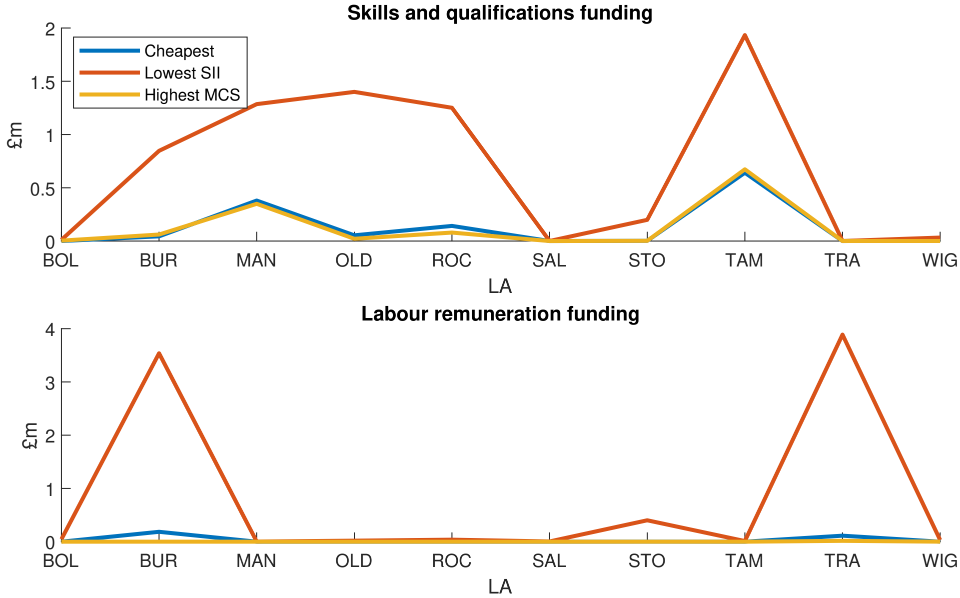


Figure 7.18: The design variable convergence for one of the solutions obtained by the NSGA-II optimizer.

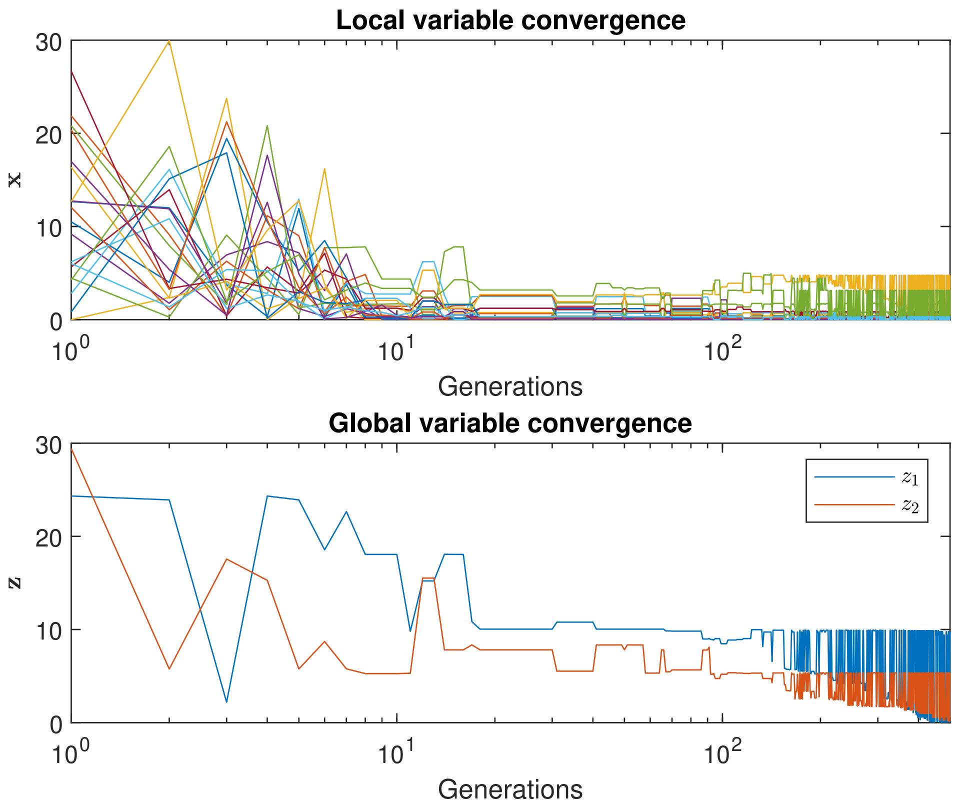
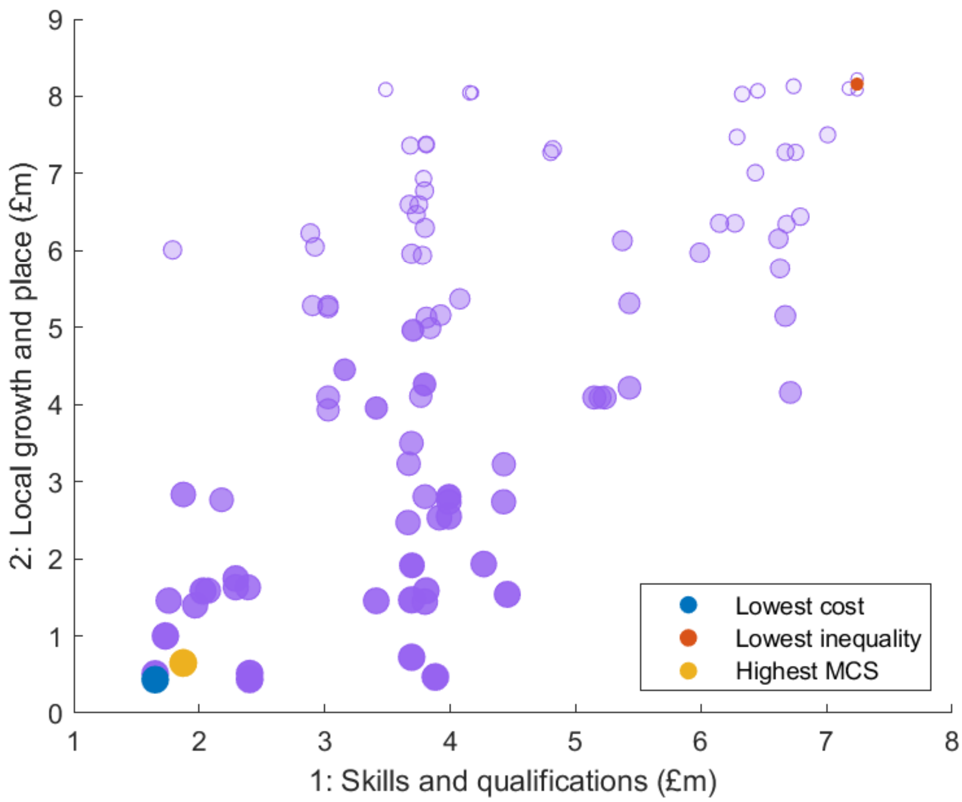


Figure 7.19: Distribution of funding by average mental health score.



Constraints

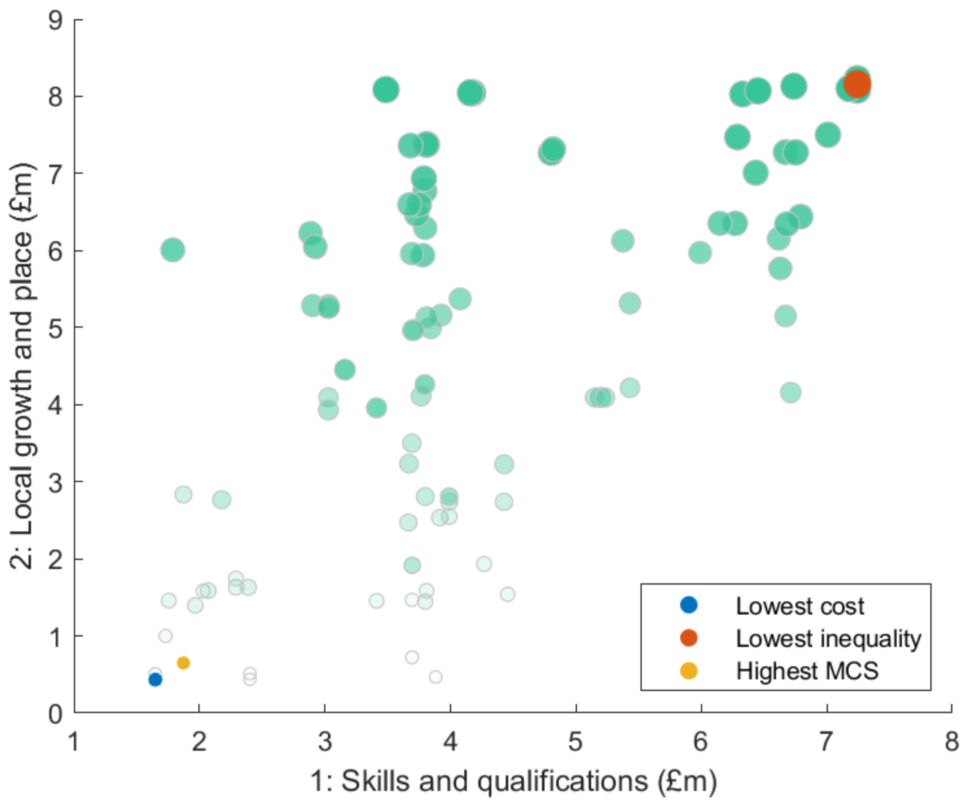
The constraint function evaluations over the 500 generations in the optimization process are shown in Figure 7.21. The horizontal axis is log scaled to highlight the constraint violation in the first ten generations.

Node responses

Figures 7.22, 7.23 and 7.24 show choropleths which each show the percentage difference from the baseline values in each of the indicator nodes (i.e. through the colour of the LA on the choropleth).

Figure 7.22 shows the percentage difference between the nodes for the solution with the highest average mental health, and the baseline. It can be seen that the changes in each of the LAs is minimal, on the magnitude of 0.1% to 10^{-4}%. Employment rate, disposable household income, and skills and qualifications univer-

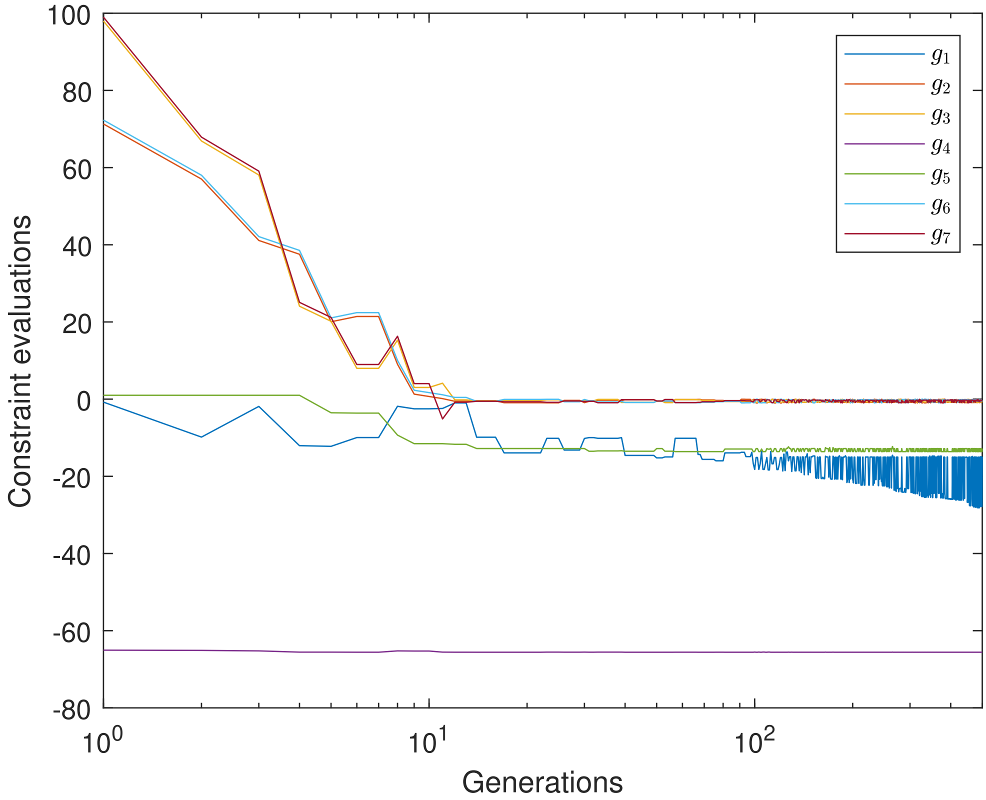
Figure 7.20: Distribution of funding by mental health inequality.



sally increase, while poverty, mental health, fuel poverty, and percentage of people on below the living wage decrease. Inequality increases most notably in Bury, but also decreases in some other LAs. Healthy life expectancy tends to vary, with some local authorities seeing an increase (Wigan, Rochdale), while others see a decrease (Stockport and Trafford).

Figure 7.23 shows the percentage difference between the nodes for the solution with the lowest mental health inequality (as measured by slope index of inequality), and the baseline. The percentage differences are much larger in this solution than in Figure 7.22. The nodes that universally increase are disposable household income, premature mortality rate, and skills and qualifications. The nodes that universally decrease are poverty, fuel poverty, healthy life expectancy and percentage of people on below the living wage. The drop in percentage of people on below the living wage is most notable in Trafford, which sees a nearly 30% drop in this node. Premature mortality rate also increases the most in Trafford, with nearly a 0.2%

Figure 7.21: The constraint function evaluations over the 500 generations in the optimization process.



increase. While child poverty and fuel poverty decrease in all LAs, they decrease the most in Trafford, Wigan and Bury.

Figure 7.24 shows the percentage difference between the nodes for the solution with the lowest cost and the baseline. The changes from baseline in these nodes are typically smaller than those in Figure 7.23, but some nodes such as child poverty, domestic household income, and fuel poverty are slightly larger. Child poverty, fuel poverty, mental health, healthy life expectancy and percentage of people on below the real living wage all decrease in this solution, with Trafford seeing the largest decreases. Premature mortality also increases the most in Trafford. Employment rate, disposable household income, premature mortality rate, and skills and qualifications increase in all LAs.

Figure 7.22: The choropleth showing the node (indicator) outputs for the solution with the highest average mental health over GMCA.

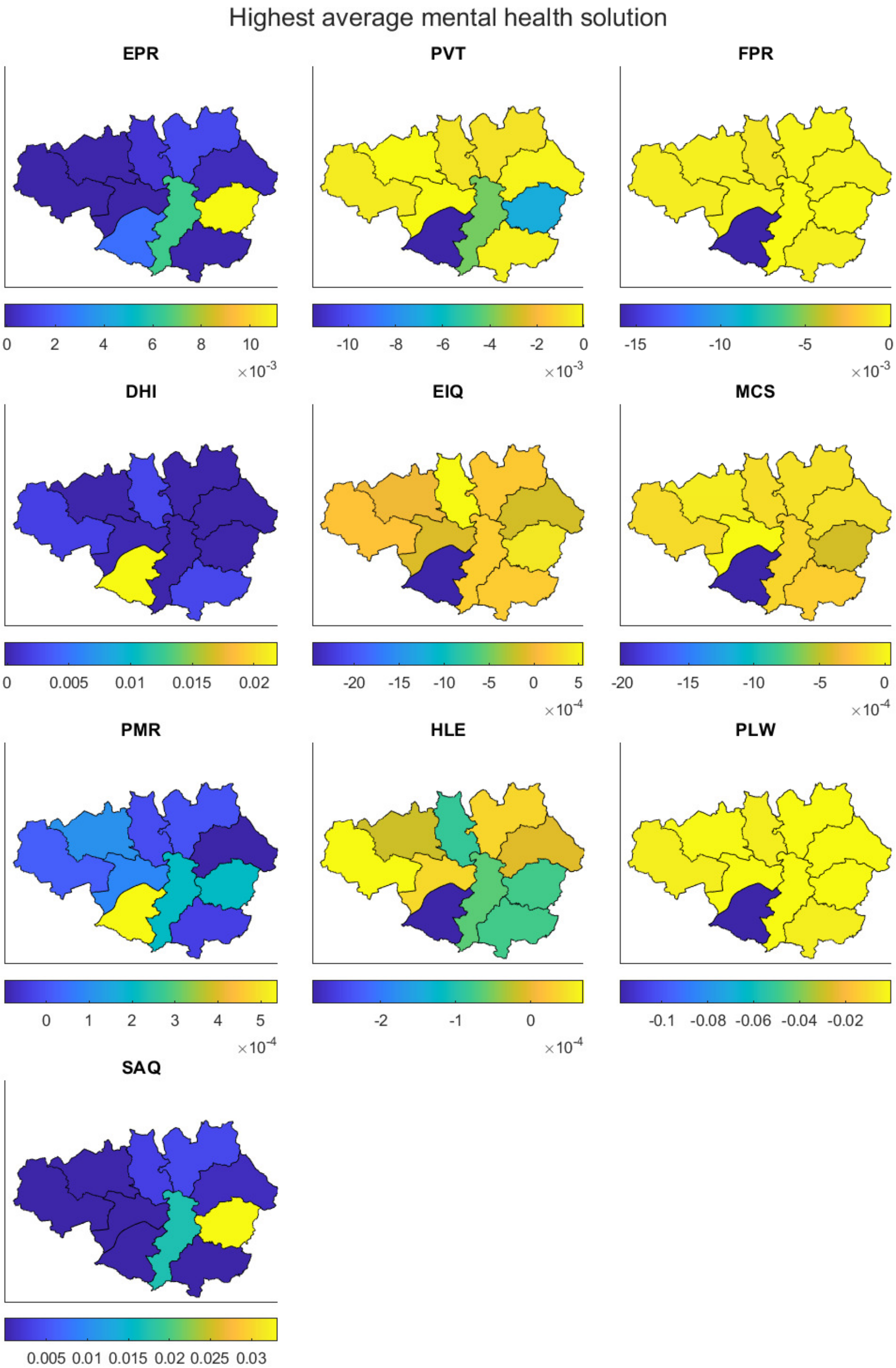


Figure 7.23: The choropleth showing the node (indicator) outputs for the solution with the lowest mental health inequality over GMCA.

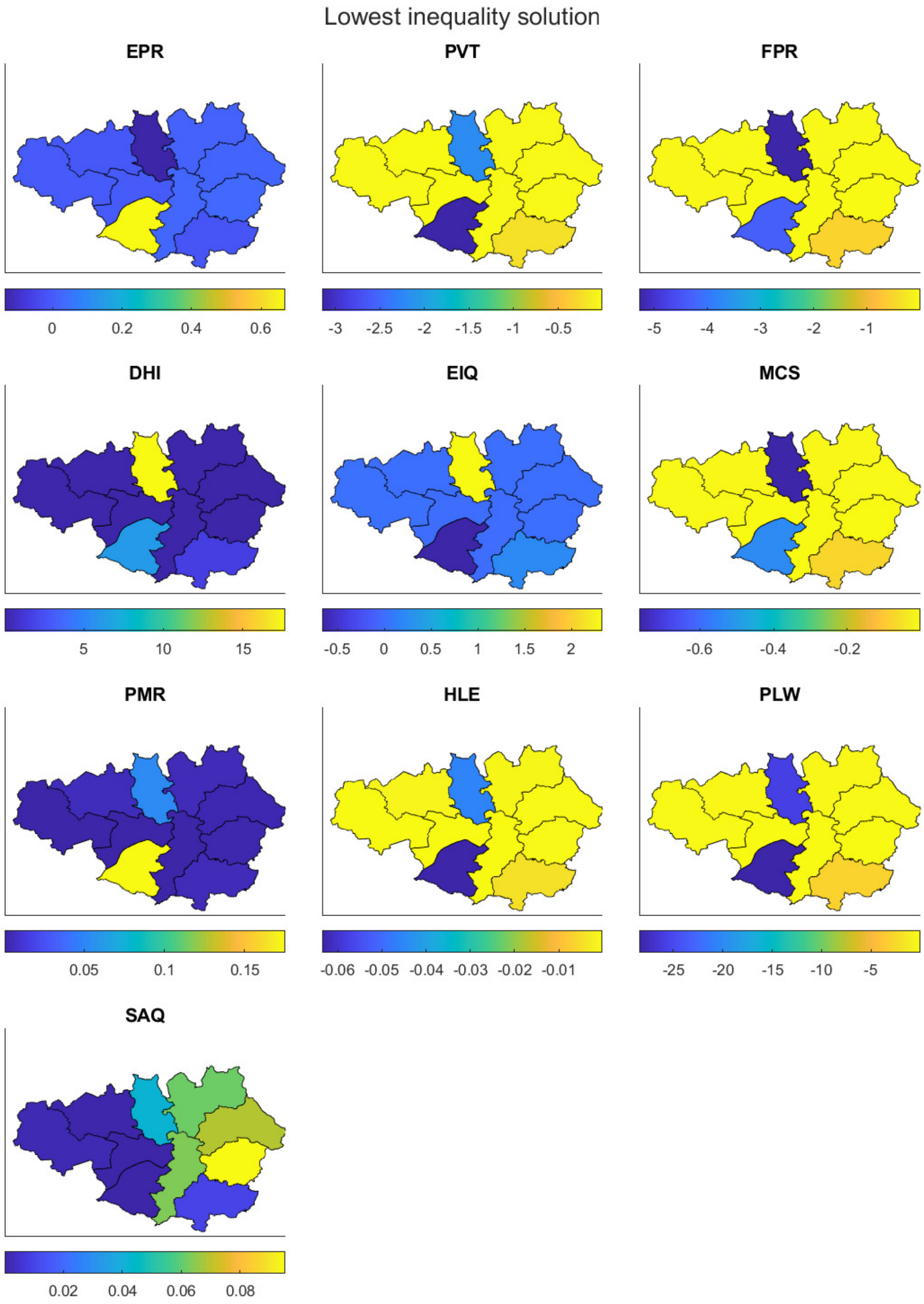
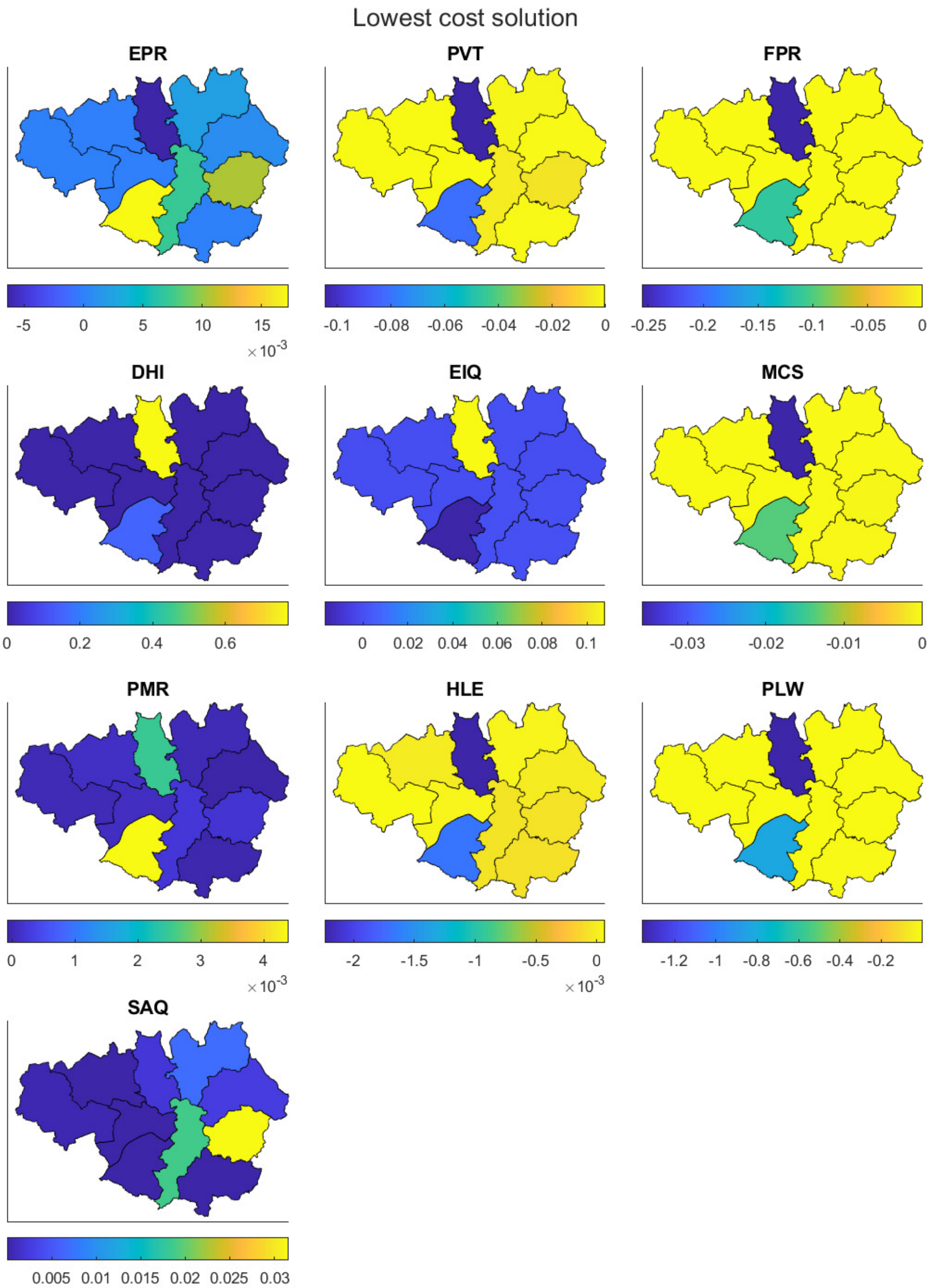


Figure 7.24: The choropleth showing the node (indicator) outputs for the solution with the lowest cost over GMCA.



7.7.3 On the slope index of inequality

As stated before, any increase in funding in the model (and by extension, an increase in skills and qualifications, or a decrease in percentage of people on below the living wage or people in non-permanent employment) will lead the SF-12 mental health scores to decrease in all clusters. The second objective in the problem relies on the slope index of inequality to reduce the inequality between the different LAs. Because of these two facts, the optimizer seeks to reduce the mental health values of the LAs with the largest baseline mental health scores. Per Figures 7.14 and 7.13, these are Trafford, Wigan, Bury, Tameside and Stockport. This is why, in the 'lowest SII' solution, the most funding goes to Trafford, Wigan and Bury in the funding for labour remuneration (see Figure 7.17).

In future iterations of this problem, it would be a good idea to either modify the second objective (minimising health inequality across GMCA), introduce a new objective, or introduce constraints to prevent this effect from occurring.

7.8 Summary and conclusion

In this chapter, an example problem that might be faced by public health policymakers in a combined local authority was introduced. This problem had three objectives, twenty disciplines and seven constraints. The multi-objective problem was solved using the MDF architecture and the NSGA-II optimizer, and the Newton solver was used for the disciplinary MDAs. The results of the optimization were analysed, including the final design variables for multiple solutions (which are important to guide decision-makers in public health), design variable convergence, objective function evaluations, and output node values for each local authority.

The results of the optimization process clearly pointed towards a dichotomy between solutions with a low inequality/high cost and a high average mental health score/low cost. The reason for this is that, in the model, investing more in skills and qualifications (discipline one) and labour remuneration (discipline two) leads to a drop in average mental health scores. However, because of this, the optimizer was able to change the mental health scores of different LAs to different degrees, meaning it was able to reduce the modulus of the slope index of inequality to near-zero.

Three key solutions were identified and analysed. These solutions were compared to each other for their objective evaluations, funding (design variable) distributions, and the corresponding node outputs in the system map. They were also compared to the baseline node values. It was found that the changes from the baseline values for the high mental health solution were very small, while the changes from the baseline values in the low inequality solutions were much larger and more noticeable.

The problem in this chapter is one of the first times the MO-MDO framework has been applied to a public policy background. Background information, data, and expert opinion were key in developing the problem formulation, from the connection between funding and skills increases, to the preferred methods for expressing inequality.

However, there are some limitations in the problem presented in this chapter. Firstly, the optimization problem is separable in disciplines (LAs), meaning it poses less of a challenge to the optimizer and the solver than it could do. Secondly, the Pareto front for the problem is unknown, meaning it is hard to evaluate and analyse the optimization solutions in an objective way, especially when compared to the results produced by a test set with a known Pareto front, such as ZDT [47] or DTLZ [88].

This chapter has shown that it is not only possible to apply an MO-MDO framework in a public health policy setting, but it is highly beneficial for organising, understanding, and appropriately representing the problem. The disciplinary partitions are useful for representing both the different LAs and the destination of the funding in each LA. Using MO-MDO methods and strategies could, in the future, be a useful tool in devolved authorities.

Chapter 8

Conclusions and future work

In this chapter, a summary and the main conclusions of this thesis are presented and discussed in Section 8.1. In Section 8.2, suggestions for future work and further research directions are given.

8.1 Summary

Multi-objective multidisciplinary design optimization (MO-MDO) is a useful branch of optimization research and practice that exploits the structure of human-made structures, products, and organisations to separate the problem out into disciplines. In this thesis, the focuses have been on, firstly, developing standardised academic scalable benchmarks and problems for use in MO-MDO research, and secondly, using an MO-MDO framework to solve a public health optimisation problem faced by devolved authorities in the UK.

In Chapter 2, the key concepts for the thesis were introduced, such as concepts in MDO, MOPs, as well as design of experiments, game theory, and Pareto dominance.

In Chapter 3, a literature review of the methods, strategies and architectures in multidisciplinary feasible (MDF) and collaborative optimisation (CO) MO-MDO was provided. These included a priori and a posteriori methods, as well as a range of novel approaches, such as linear physical programming, fuzzy approaches, and incorporation of genetic algorithm concepts into the MO-MDO structure.

In Chapter 4, a scalable MO-MDO test problem based on the non-MDO ZDT1 problem was presented and then solved using the MDF MDO architecture. This test problem is easily scalable in number of design variables and disciplines, meaning it can be used to analyse the effect of increasing the size of the problem. The number of disciplines in the problem presented in Chapter 4 was varied between two and fourteen and the number of variables was correspondingly varied between 30 and 150. This was compared with the results from the non-MDO ZDT1 problem and it was found that the interactions between disciplines caused a larger error between the solution set and the Pareto front, than simply increasing the number of design variables in the non-MDO ZDT1 problem.

In Chapter 5, the test problem framework introduced in Chapter 4 was extended to the rest of the continuous problems in the ZDT test set. The linking relationships (also referred to as topologies) between the disciplines were varied from that presented in Chapter 4, with more complex topologies taking more of an important role. These included two-in-two-out (TITO) and all-in-all-out (AIAO) structures. In addition to this, the linking variables (and corresponding optimal values) were incorporated into the problem by introducing a penalty for deviation from the optimal solutions. It was found that the MO-MDO framework particularly struggled with the MDO-ZDT problem, especially when there were large numbers of disciplines.

In Chapter 6, the MDO-ZDT test problems were extended further for use in a distributed architecture. This architecture was CO. Constraints were used in the disciplinary sub-problems, and designed in such a way that when these constraints were satisfied, it would be possible to see it in the solution set. The relaxation factor for the system-level consistency constraint was varied. It was found that increasing the number of disciplines from two to three reduced the spread of the solution set, and deviated from the shape of the Pareto front of the continuous non-MDO ZDT problems. It was also found that, as the relaxation factor became smaller, it became harder to meet the termination criteria for optimization.

In Chapter 7, an MO-MDO problem was developed that was suitable for informing devolved government policies in public health and health inequality. The MO-MDO framework was implemented with the intention of solving this problem. The problem contained three objectives, a large number of disciplines (twenty) and seven constraints. Three key solutions were identified and analysed, and then compared with each other.

8.2 Future work

Extension to more advanced MO-MDO test problems

One of the key considerations for future work in the field of MO-MDO should be developing more advanced MO-MDO test problems. As noted in [89], the ZDT test suite (as well as some of the other popular test sets in MOP research such as DTLZ) is missing several important features for effective test suites, such as non-separability, dissimilar ranges and domains, and varying (but known) Pareto front geometries. However, it should also be noted that it is difficult to build MO-MDO test suites based on some of the other, more complex MOP test suites for several reasons: BBOB-biobj [91] only contains bi-objective test problems, and the Pareto fronts are unknown; there is also no clear way of introducing the three types of variables in the WFG problems [89]. It is necessary for the analysis of MO-MDO algorithms that the original Pareto front should be recovered, and local, linking and global variables should be used, so these other test suites do not satisfy the needs of an MO-MDO test suite.

It is important to note that it is not necessarily the case that these more advanced test sets should be derived from existing MOPs. While it may be more straightforward to do so in some cases, it would also be beneficial to develop a test set that is customised for MO-MDO strategies and frameworks. In addition to this, some of the best MOP benchmarks are difficult to adapt to MO-MDO frameworks in an obvious manner due to the very features that make them good benchmarks, such as non-separability.

One way of approaching this could be to identify features of problems that would be useful in MO-MDO research, such as those identified by Huband et al. [89]. On top of the recommendations for MOPs, features such as varied disciplinary linkages, scalable objectives and disciplines, and adaptions for use in other architectures should be considered.

Introduction of interactive methods to MO-MDO

In the review of existing MO-MDO methods presented in Chapter 3, two of the main approaches to handling operator preferability in MOPs are identified in MO-MDO literature. These are the a priori and the a posteriori methods, in which the operator or decision-maker's preferences are incorporated either before or after the optimization process, respectively. However, it was identified that the interactive methods in MOPs are not exploited.

Interactive methods are hugely important in MOP practise and research because they allow the decision-maker to involve themselves during the optimization process. In fact, it is argued that interactive methods allow the decision-maker to learn about the problem and what is considered possible within the solutions, and then make the decisions while utilising the knowledge built during the 'learning' phase [168].

The sort of advantages gained by these interactive processes could be very useful in MO-MDO approaches, as they are in non-MDO MOPs. Some of the future work in the field of MO-MDO should focus on developing a variety of interactive methods (as it is also noted in [168] that different interactive methods may better or worse suit different decision-makers).

Visualisation of MO-MDO results

While many of the existing MOP visualisation tools are applicable to solutions in MO-MDO problems, it would be useful to develop a standardised visualisation strategy for clarity and ease of comparison.

There are two motivating factors for this:

1. In MDF problems (both monolithic and distributed), a solver is used to converge a multidisciplinary analysis (MDA) to enforce consistency between linking variables. This may take several, if not dozens, hundreds, or even thousands, of iterations to converge. These MDAs each happen between iterations of the optimizer at the system level. In other words, for every one objective function evaluation in the optimizer, there may be multiple iterations of the solver occurring.

2. In distributed MO-MDO problems, the same problem occurs with the optimizers at the subsystem levels. Each iteration of the optimizer at the system level means several other optimizers are running at the same time.

Currently, there is no easy method for handling and visualising the data obtained in the two scenarios detailed above. Future work should consider these points.

Performance indicators in distributed MO-MDO

In a similar vein to the above, performance indicators specialised for use in MO-MDO are not present in the literature. While there are many performance indicators in non-MDO MOP research, there are none for specialised use in MO-MDO. It should be possible to use these performance indicators in MDO architectures where there are several optimizers working at once, such as in collaborative optimization (CO).

Comparison of different optimizers in MO-MDO

In this thesis, the NSGA-II optimizer [111] has been used in all studies. While there are ample studies in the MOP literature comparing the performance of different multi-objective optimizers, there is a lack of literature in the field of MO-MDO doing the same, especially for distributed architectures where more than one optimizer is working at once. The identification of the most effective optimizers in different MO-MDO settings would be a useful area of further research.

Improvement of convergence in multi-objective CO in problems with many disciplines

In Chapter 6, a multi-objective CO architecture was used to solve the MDO-ZDT problems in Chapters 4 and 5, with additional constraint functions. It was found that the system-level optimizer struggled to converge a solution the most when there were more than two disciplines. Finding a strategy for improving these convergence characteristics should be considered a priority in future distributed MO-MDO research.

Implement larger and more complex problems, including uncertainty

While the case study in Chapter 7 produced positive results at a local authority level in a devolved authority in England, the problem was highly granular. Increasing granularity in this problem could involve reducing the geographical areas in the problem to middle layer super output areas (MSOA) or lower layer super output areas (LSOA) or even at the individual level. However, this could mean that the number of disciplines, design variables, or linking variables are vastly increased. Additionally, it would mean that the number and difficulty of MDAs would also increase.

While some of this thesis has focused on scalability in MO-MDO, there is scope to increase it even further. This avenue of future work would involve much further thought regarding storage and transfer of data between disciplines, use of metamodels, or even slight adjustments in the MO-MDO architecture used. However, the results would be useful for the devolved authorities working on these types of problem.

If these types of problems are to be pursued, the inclusion of robust MO-MDO methods should be considered. Uncertainty is guaranteed in the sort of problems faced by devolved authorities, and should therefore be included in the problem and be part of the solution. This may involve using the robust methods identified in Chapter 3, the review of relevant literature, or developing different strategies for incorporating uncertainty in large MO-MDO problems.

Implement more real-world problems in MO-MDO

While there are some real-world and real-world-inspired problems in the field of MO-MDO, there is a need to increase the number of these problems to further investigate and demonstrate the usefulness of these strategies. Real-world problems are likely to include structures and features that are difficult for convergence in multi-objective optimizers, such as complex disciplinary analyses (and may require the use of meta-models in some cases), many constraints and objectives.

Modelling and analysis of MO-MDO algorithms

One typically understudied area in MO-MDO is modelling of the architectures to analyse convergence properties or proof of correctness. There is some evidence of this in some MDO architectures, particularly CO [160], but these analyses are missing from the MO-MDO literature in large. Some of the future work in this area should be dedicated to analysis of MO-MDO algorithms, including finding algorithms that guarantee convergence. In support of this point, it has been noted in [41] that some of the MO-MDO approaches at the time of writing (2017) do not include proofs of correctness.

Appendix A

Relevant MDO architectures

In this Appendix, the architectures for four key MDO architectures are included, along with their formulation. Architectures are adapted from [24], and see the same source for more in-depth explanations of each architecture.

A.1 Multdisciplinary feasible (MDF)

MDF is a monolithic architecture, meaning the whole optimisation problem is on one level. Feasibility is prioritised before convergence of the objective function. All three types of variables (linking, local, global) are used in the objective and/or the constraint functions.

The system-level problem is expressed as

$$\begin{aligned} \text{min} \quad & f(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathbf{z}) \\ \text{w.r.t} \quad & \mathbf{x}, \mathbf{z} \\ \text{s.t.} \quad & \mathbf{c_0}(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathbf{z}) \leq 0 \\ & \mathbf{c_i}(\mathbf{x}_i, \mathbf{y}_i(\mathbf{x}_i, \mathbf{y}_{j \neq i}, \mathbf{z}), \mathbf{z}) \leq 0, \qquad i = 1, ..., N \end{aligned} \tag{A.1}$$

A.2 Collaborative optimization (CO)

CO is a distributed architecture which contains a consistency constraint at the system level and consistency objectives at the subsystem level. Copy variables are used to maintain subsystem independence.

In CO, the system problem is expressed as

$$\begin{aligned}
\text{min} \quad & f(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \mathbf{z}) \\
\text{w.r.t} \quad & \hat{\mathbf{x}}, \hat{\mathbf{y}}, \mathbf{z} \\
\text{s.t.} \quad & g(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \mathbf{z}) \leq 0 \\
& J^*(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \mathbf{z}) = \sum_{i=1}^{N} \sum_{j=1}^{n_z} (z_j - \hat{z}_{i,j})^2 + \sum_{i=1}^{N} \sum_{j=1}^{n_{x_i}} (\hat{x}_{i,j} - x_{i,j})^2 + \\
& \sum_{i=1}^{N} \sum_{j=1}^{n_{y_i}} (\hat{y}_{i,j} - y_{i,j})^2
\end{aligned} \tag{A.2}$$

And the subsystem-level problem k is expressed as

$$\begin{aligned}
\text{min } J_k(\mathbf{x}_k, \mathbf{y}_k, \hat{\mathbf{z}}_k) = \sum_{i=1}^{n_z} (z_i - \hat{z}_{k,i})^2 + \sum_{i=1}^{n_{x_k}} (\hat{x}_{k,i} - x_{k,i})^2 + \\
\sum_{i=1}^{n_{y_k}} (\hat{y}_{k,i} - y_{k,i})^2
\end{aligned} \tag{A.3}$$

$$\text{w.r.t } \mathbf{x_k}, \mathbf{y_k}, \hat{\mathbf{z}}_\mathbf{k}$$

$$\text{s.t. } g_k(\mathbf{x}_k, \mathbf{y}_k, \hat{\mathbf{z}}_k) \leq 0$$

A.3 Concurrent subspace optimization (CSSO)

In CSSO, the subsystems are not only responsible for their own constraints, but the constraints of the system-level problem and the constraints of the other subsystems as well. The system objective function is also shared between all subsystems. In this particular formulation of CSSO, metamodels of the linking variables from other subsystems (notated $\tilde{y}$) are used to replace the linking variables in subsystem i from other subsystems j.

In CSSO, the system problem is expressed as

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}}(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}})) \\ \text{w.r.t} \quad & \mathbf{x}, \mathbf{z} \\ \text{s.t.} \quad & g(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}}(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}})) \leq 0 \\ & g_i(\mathbf{x}_i, \tilde{\mathbf{y}}_i(\mathbf{x}_i, \tilde{\mathbf{y}}_{j \neq i}, \mathbf{z}), \mathbf{z}) \leq 0 \end{aligned} \tag{A.4}$$

And the subsystem-level problem i is expressed as

$$\begin{aligned} \min \quad & f(\mathbf{x}, \mathbf{z}, \mathbf{y}(\mathbf{x}, \tilde{\mathbf{y}}_{j \neq i})) \\ \text{w.r.t} \quad & \mathbf{x}_i, \mathbf{z} \\ \text{s.t.} \quad & g(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}}(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}})) \leq 0 \\ & g_i(\mathbf{x}, \mathbf{z}, \mathbf{y}(\mathbf{x}, \mathbf{z}, \tilde{\mathbf{y}}_{j \neq i})) \leq 0 \\ & g_j(\mathbf{z}, \tilde{\mathbf{y}}_j(\mathbf{z}, \tilde{\mathbf{y}})) \leq 0 \end{aligned} \tag{A.5}$$

A.4 Analytical target cascading (ATC)

ATC is a penalty method. The penalty functions are expressed as $\Phi(\bullet)$ and are used to penalise differences between the 'active' variables and the copy variables (similar to those that are found in CO). Additionally, at the subsystem level, the system objective is summed into the subsystem objective as well as the relevant penalty functions, meaning that the system objective is also minimised at the subsystem level.

The system-level problem for ATC is

$$\begin{aligned} \min \quad & f(\mathbf{x}, \hat{\mathbf{y}}) + \sum_{i=1}^{N} \Phi_i(\hat{\mathbf{z}}_i - \mathbf{z}, \hat{\mathbf{y}}_i - \mathbf{y}_i(\mathbf{x}_i, \mathbf{y}, \mathbf{z})) + \Phi(g(\mathbf{x}, \hat{\mathbf{y}})) \\ \text{w.r.t} \quad & \mathbf{z}, \hat{\mathbf{y}} \end{aligned} \tag{A.6}$$

and the subsystem-level problem is

$$
\begin{aligned}
\text{min} \quad & f(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_{j\neq i}), \mathbf{y}_{j\neq i}) + f_i(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_{j\neq i}), \mathbf{y}_{j\neq i}) + \\
& \Phi_i(\hat{\mathbf{y}}_i - \mathbf{y}_i(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_{j\neq i})) + \Phi(g(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_{j\neq i}), \mathbf{y}_{j\neq i})) \\
\text{w.r.t} \quad & \hat{\mathbf{z}}, \mathbf{x}_i \\
\text{s.t.} \quad & g(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_i(\hat{\mathbf{z}}_i, \mathbf{x}_i, \mathbf{y}_{j\neq i})) \leq 0
\end{aligned}
\tag{A.7}
$$

Appendix B

Concurrent subspace optimization and analytical target cascading MO-MDO approaches

In this Appendix, the concurrent subspace optimization (CSSO) and analytical target cascading (ATC) methods excluded from the literature review in Chapter 3 are discussed. These methods are in the appendix because they are not directly relevant to the rest of the research in this thesis.

B.1 A priori approaches

Summed objective ATC

In the multi-objective ATC study [118] using the formulation from [169], an aircraft system is optimised, whose partitioning is sequential-based (with subsystems based around the cruise, take-off and second segment climb regimes). The problem contains four summed objectives which are not weighted, however the penalty weights associated with the ATC formulation were decided via trial and error. This formulation is different to many of the other formulations discussed in this section, because it is tri-level (compared to the bi-level formulations popular in MDO research). The outcomes of the ATC formulation were compared with the outcomes of the same problem with an AAO implementation. It was found that the ATC formulation took longer to run than the AAO formulation, but provided more flexibility and in some cases, the optimisation results were better (more minimised) than those in AAO.

ATC-PP and ATC-LPP

Wang et al. proposed a multi-objective ATC formulation by combining this architecture with linear physical programming [170]. Preferences are applied at the system level, are usually articulated before the optimisation process is carried out, and the subsystem levels solve a consistency problem. The system-level problem is multi-objective, while the subsystem level problem seeks to minimise the difference between objective function obtained by subsystem i and the corresponding objective obtained by the system-level optimiser f_i^U. The system-level problem is

$$\begin{aligned} \min \quad & J = [J_1, J_2, J_3, J_4] \\ \text{s.t.} \quad & ||R_{sub} - R_{sub}^L||_2^2 \leq \varepsilon_R \\ & ||y_{sub} - y_{sub}^L||_2^2 \leq \varepsilon_y \\ & g_{sys}(R_{sub}, \tilde{x}_{sys}) \leq 0 \\ & h_{sys}(R_{sub}, \tilde{x}_{sys}) = 0 \\ & \tilde{x}_{sys}^{min} \leq \tilde{x}_{sys} \leq \tilde{x}_{sys}^{max} \\ \text{where} \quad & J_k = \sum_{i=1}^{n} w_{i,k} \left(f_i(x) - d_{i,k} \right) + \varepsilon_R + \varepsilon_y \end{aligned} \tag{B.1}$$

and the subsystem-level problem is

$$\begin{aligned} \min \quad & ||f_i(x) - f_i^U(x)|| \\ \text{s.t.} \quad & g_i(x) \end{aligned} \tag{B.2}$$

where n is the number of system goals of soft classes, g_{sys} and h_{sys} are the system-level inequality and equality constraints respectively, $w_{i,k}$ are weights expressed by the designer, J_k is the function of deviation variables at soft class k, J

is the aggregate preference function.

B.2 A posteriori approaches

B.2.1 Analytical target cascading with solution sets

The authors in [171] proposed a modification to classical non-hierarchical ATC by adding a weighted sum component to each of the subsystems. As this architecture is non-hierarchical, there is no system-level problem. The subsystem i problems are formulated as

$$\begin{aligned} \min \quad & \omega_i f_i + \sum_{i=1, j\neq i}^{n_s} \phi(\mathbf{y_j} - \mathbf{y_i}) \\ \text{s.t.} \quad & \mathbf{g_i} \leq \mathbf{0} \\ \text{where} \quad & \phi(\mathbf{y_j} - \mathbf{y_i}) = \mathbf{v_{j,i}}(\mathbf{y_j} - \mathbf{y_i})^T + ||\omega_{\mathbf{j,i}} \circ (\mathbf{y_j} - \mathbf{y_i})||_2^2 \end{aligned} \tag{B.3}$$

where $\mathbf{v_{j,i}}$ are Lagrange multipliers and $\omega_{\mathbf{j,i}}$ are penalty weights; this approach is adapted from [112] and more information on the update method can be found in [70]. The Pareto set of the problem, made up from the subsystem objectives, is obtained by varying the penalty weights $\omega_{\mathbf{j,i}}$.

B.2.2 CSSO-style methods

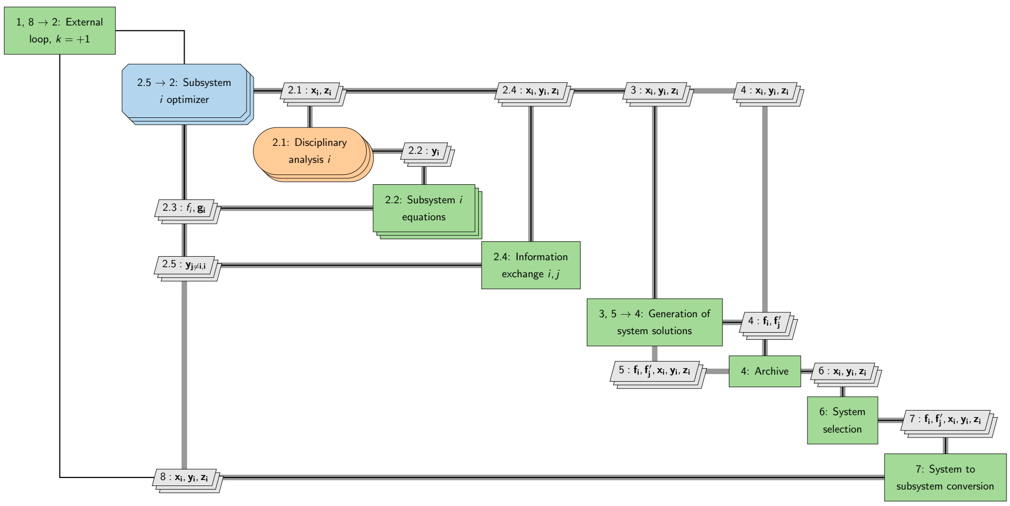
The CSSO-style methods detailed in this subsection can be sorted into four main variants/strategies:

- Strategies using techniques from outside the non-MDO MOP research, such as metamodels or neural networks.
- Robust strategies, where the optimisation is robust to small variations in the global variables.
- MOP-based methods, using methods common in non-MDO MOP research.
- Adaptive weighted sum strategies.

Duro et al. distributed design

Duro et al proposed a five step approach to distributed optimisation of complex products, with optimisers at the subsystem level, and a selection algorithm operating at the system level [13]. The subsystems begin by optimising their individual problems, then send the solutions for exchange, where a supervised learning model (in this case, an artificial neural network) generates a predicted set of linking variables for each subsystem. These predictions are further used in the optimisation for each subsystem. Then, system solutions are obtained by combining the solutions for each subsystem, and then all solutions are archived. A set of system solutions are selected from the archive based on a relaxed consistency constraint, and these solutions are fed back into the subsystem problems, replacing some fraction of the initial population. This approach shows similarities to the MORDACE approach. In both cases, a co-ordinator is employed at the system level rather than an optimiser – the best solutions are chosen by the co-ordinator and sent back to the subsystems rather than used in their own optimisation problems. This approach is similar to CSSO, which is also a multi-level approach with a co-ordinator at the system level.

Figure B.1: The XDSM diagram of Duro et al.'s method adapted from [13].



Multidisciplinary optimisation and robust design approaches applied to concurrent engineering (MORDACE)

MORDACE [14] is a CSSO-style method where the subsystems solve a multi-objective problem and the system level finds a compromise between the sets of design variables produced by the subsystems.

The problem solved by the subsystem optimisers is

$$\begin{aligned} \min \quad & f_i\left(\mathbf{x_i}, \mathbf{z_i}, \tilde{\mathbf{y}}_{\mathbf{j}\neq\mathbf{i}}(\mathbf{x_i}, \mathbf{z_i})\right), \sigma_{T,i} \\ \text{s.t.} \quad & g_i\left(\mathbf{x_i}, \mathbf{z_i}, \tilde{\mathbf{y}}_{\mathbf{j}\neq\mathbf{i}}(\mathbf{x_i}, \mathbf{z_i})\right) \\ \text{w.r.t} \quad & \mathbf{x_i}, \mathbf{z_i} \end{aligned} \tag{B.4}$$

where $\tilde{y}_{j\neq i}$ is the approximated linking variable from subsystem j, calculated using a simplified model of discipline j, and $\sigma_{T,i}$ are the results of the sensitivity analysis calculated within the subsystem analysis, and can be seen in the XDSM of the MORDACE method in Figure B.2. Note that subsystem i does not create linking variable $\mathbf{y}_i$, as the other subsystems j create their own approximated version of the corresponding linking variable. Additionally, the problem solved by the subsystems is a multi-objective problem, aiming to minimise both the subsystem objective f_i and the sensitivity analysis. This ensures the variables are robust to small changes in the global variables, and allows the compromise search to proceed without having small changes in $\mathbf{z}_{j\neq i}$ disrupt the performance of the system. Also note that the global variables have the subscript i – what this means is that each subsystem finds solutions with its own values of $\mathbf{z}_i$, without taking consistency between subsystems into account. Consistency between all $\mathbf{z}_i$ is obtained at the system level.

The system-level problem is a compromise search, where pairs of solutions k and l for each of the subsystems respectively, are evaluated using the weighted Euclidean distance between the global variables of each solution that Pareto-dominates the "original" solution. To obtain the weights, another sensitivity analysis is carried out (see [172] for more details on robust design in multidisciplinary design optimisation). In [14], this is a t-student test, and indicates the strength of the relationship between the pairs of global variables. Solutions with high robustness to variable error and a small distance between pairs are carried forwards. For these pairs, their weighted mean is considered the solution. These are validated before solution acceptance.

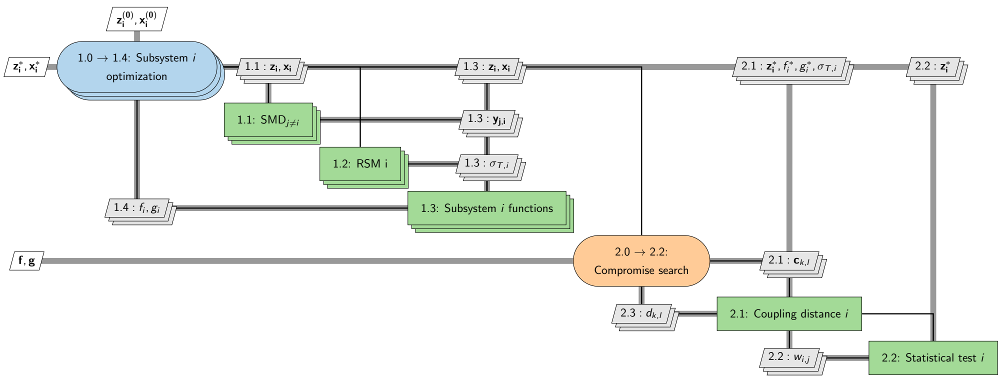
The authors note that the number of design pairs has to be limited by the user to reduce computational expense to a manageable level. Additionally, it should be noted that adding extra disciplines increases the dimensionality of the problem exponentially, making the use of this method with many-discipline problems prohibitively time-consuming.

While the system-level problem is termed a compromise search in [14], it could also be considered a combinatorial optimisation problem of the form

$$\begin{aligned} \min \quad & d_{k,l}(\mathbf{z_k}, \mathbf{z_l}),\ \sigma_T \\ \text{where} \quad & \mathbf{z_k} \in Z_k, \quad \mathbf{z_l} \in Z_l \end{aligned} \tag{B.5}$$

where Z_k is the set of global variables resulting from discipline k and Z_l is the set of global variables resulting from discipline l. σ_T is the outcome of the sensitivity analysis, also to be minimised.

Figure B.2: The XDSM diagram of the MORDACE method adapted from [14].



Multi-Objective Pareto Concurrent Subspace Optimization (MOPCSSO)

MOPCSSO [7, 173, 174, 175] was developed in response to the earlier, a priori methods with the intention of addressing some of the problems in these architectures identified by the authors. These were that the a priori methods did not properly deal with highly coupled problems, or problems with “separate but coupled disciplinary groups” [7]. Additionally, the aim was to avoid the issues inherent with methods such as weighted sums like requiring the priori knowledge to identify appropriate weights for the objective functions, and not producing a set of solutions that could be evaluated by a human designer following the optimisation process.

MOPCSSO is presented primarily as a one-discipline-one-objective architecture, but it is possible to have multiple objectives in one subproblem.

As in classical CSSO, the subsystems in MOPCSSO have a responsibility to ensure that their solutions do not produce infeasible results in other subsystems, and also that the resulting objective function values are no worse than at the starting point. This is expressed using the responsibility coefficient r_i^j; the subscript i indicates which subsystem it is ‘attached’ to (i.e. it belongs to subsystem i) and the superscript j indicates which subsystem it is responsible for.

MOPCSSO is the result of three key modifications to the classical CSSO architecture. Firstly, in classical CSSO, there is an optimal sensitivity analysis and subsequent co-ordination optimisation process that determines the values of the responsibility and trade-off coefficients. In MOPCSSO, this is replaced with a simple re-initialisation of the responsibility coefficients before the parallel subsystem optimisation processes. Design variables are therefore also reallocated at every iteration of the process, just before the re-initialisation of $\mathbf{r}$. Secondly, the parameter ρ, which identifies how much the cumulative constraints represent all constraints in the problem, is modified dynamically throughout the optimisation process. ρ is set to start from a small value and increases throughout. Thirdly, the authors state that starting from an infeasible point may lead to cumulative constraint violation, so the starting point must be feasible in the first instance.

Following this, additional changes are made to the formulation to find the extrema on the theoretical Pareto front, and also to ensure an even distribution of solutions.

The subsystem i problem is formulated as

$$\begin{aligned} \min \quad & f_i(\mathbf{x_i}) \\ \text{s.t.} \quad & C_j(\mathbf{x_i}) \leq C_j^0(\mathbf{x_i})(1 - r_i^j), \qquad j = 1, ..., N \\ & f_j(\mathbf{x_i}) \leq f_j^0 \qquad j = 1, ..., N, \ j \neq i \end{aligned} \tag{B.6}$$

where C_j is the cumulative constraint function for subsystem j, a superscript C^0 and f^0 indicate the value of the cumulative constraint function and objective function, respectively, at the beginning of the optimisation cycle – this ensures that the obtained solutions are no worse than the previous solutions.

There is no system-level objective function in MOPCSSO.

Note that, in this formulation, the subsystems rely on disjoint sets of design variables, and are rearranged between subsystems at each iteration. Therefore the only variable types that exist are local variables; however, the subsystem couplings are handled via the responsibility parameter, cumulative constraints and objective function constraints in Equation B.6.

The main limitation to this method is that the sensitivity and system analyses are computationally expensive and time-consuming, as explained in [15]. While this is acknowledged in [7], they also explain that the computational expense of these are offset by the fact that the optimum sensitivity analysis and co-ordination optimisation problem found in classical CSSO are removed.

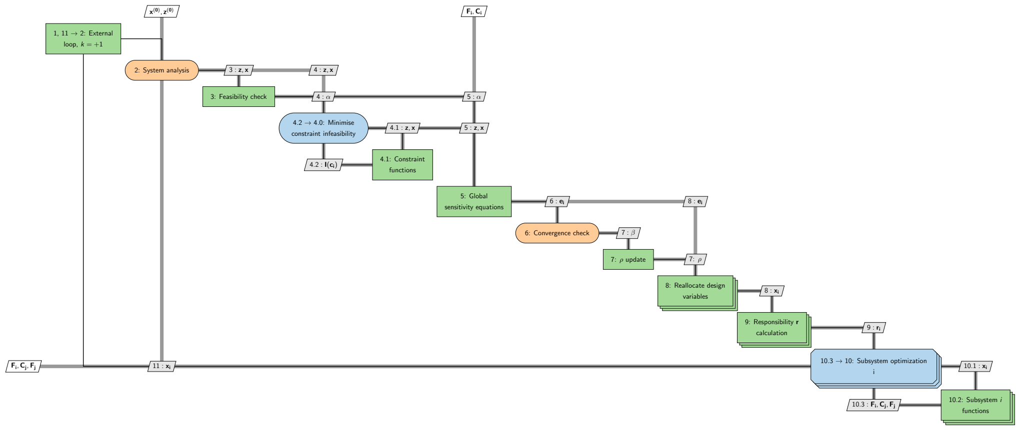
The XDSM for MOPCSSO is shown in Figure B.3.

Multi-Objective Genetic Algorithm Concurrent Subspace Optimization (MOGACSSO)

MOGACSSO is an extension of MOPCSSO. One of the main limitations of MOPCSSO is that it produces one solution at a time – MOGACSSO was developed to produce multiple Pareto solutions [15].

MOGACSSO replaces the system and sensitivity analyses found in MOPCSSO with response surface methods (in this particular instance, an artificial neural

Figure B.3: The XDSM diagram of the MOPCSSO method adapted from [7].



network (ANN)) to reduce the computational effort required to reach a solution. A small number of the initial population are used to train the ANN, from which the approximate constraint and objective function evaluations can be obtained.

Additionally, the subsystem optimisation problems are multi-objective and take the objective functions of other subsystems, transforming Equation B.6 into

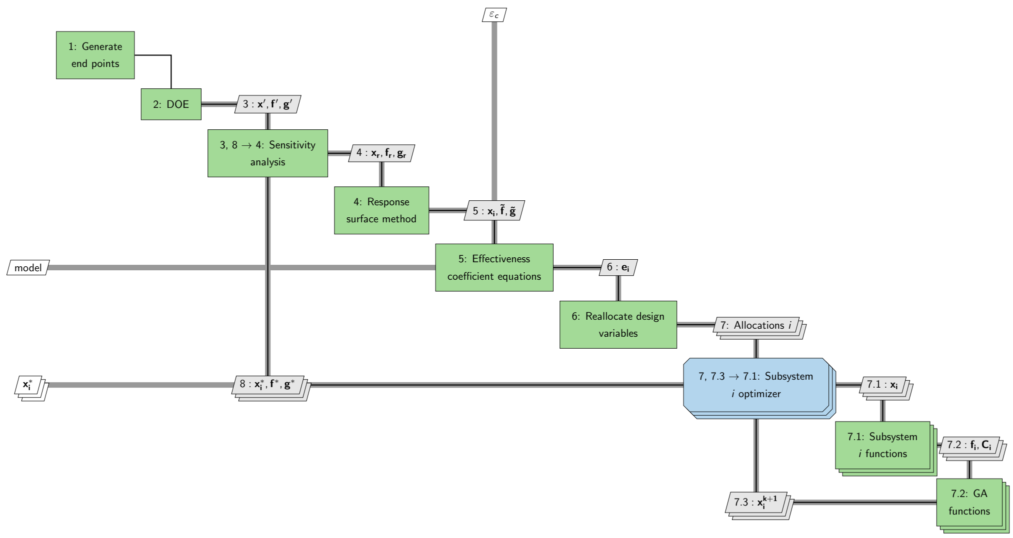
$$\begin{aligned} \min \quad & f_i(\mathbf{x_i}), f_j(\mathbf{x_i}) \\ \text{s.t.} \quad & C_j(\mathbf{x_i}) \leq C_j^0(\mathbf{x_i})(1 - r_i^j), \qquad j = 1, ..., N \\ & f_j(\mathbf{x_i}) \leq f_j^0 \qquad j = 1, ..., N, \ j \neq i \end{aligned} \tag{B.7}$$

Finally, the sensitivity analyses conducted in MOPCSSO are valid only for one point, as only one point is optimised at once. Because MOGACSSO aims to optimise a set of points at once, the equation to calculate the effectiveness coefficients are modified to take into account all subsystems.

Using a benchmark problem [176], it is shown that the MOGACSSO method is more efficient than MOPCSSO at generating a set of solutions. However, a comparison of accuracy of results is not undertaken. Additionally, scalability is not proven in the paper, with only two disciplines discussed – it is acknowledged in [15] that MDO problems are frequently very large, so it is imperative that scalability be assured for such architectures.

The XDSM for the MOGACSSO architecture is shown in Figure B.4. Note that the prime symbol ($\mathbf{x}'$ for instance) shows a subset of the design variables – in this case, it is the ones that are used to conduct the sensitivity analysis, the r subscript indicates outputs of the sensitivity analysis that have a rank between one and 10, and a tilde $\sim$ shows a model output.

Figure B.4: The XDSM diagram of the MOGACSSO method adapted from [15].



Adaptive weighted sum CSSO (AWSCSSO)

AWSCSSO was proposed in [177]. As the name implies, AWSCSSO is a combination of the adaptive weighted sum method proposed in [178], which is based on the weighted sum method (which produces a set of solutions) proposed by Das and Dennis [179], and classical CSSO. It is important to note here that 'weighted sum' does not refer to the a priori weighted sum of objectives, but is another method for Pareto front generation, and is particularly useful in cases where the Pareto front is nonconvex, demonstrated in [177] on a variation of the Osyczka and Kundu function [180]. The variation of CSSO used in AWSCSSO is the same as that proposed in [7] and visualised in Figure B.3 – these are found in the sub-optimisations.

In this approach, there are two main 'steps': firstly, the subsystems are optimised with the weighted sum objective, where the objectives from other subsystems are approximated. This optimisation problem i is expressed as

$$\begin{aligned}
\min \quad & \omega_i f_i(\mathbf{x}_i, \mathbf{y}_i, \tilde{\mathbf{y}}_j) + \omega_j \tilde{f}_j(\mathbf{x}_i), \qquad j \neq i \\
\text{s.t.} \quad & C_i(\mathbf{x}_i, \mathbf{y}_i, \tilde{\mathbf{y}}_j) \leq C_i^0(1 - r_i^i) \\
& \tilde{C}_j(\mathbf{x}_i) \leq C_j^0(1 - r_i^j) \\
& \mathbf{g_i}(\mathbf{x}_i, \mathbf{y}_i, \tilde{\mathbf{y}}_j) \leq 0
\end{aligned} \tag{B.8}$$

where, as in MOPCSSO [7] and MOGACSSO [15], C_i is the cumulative constraints of all constraints in the non-MDO problem and a tilde indicates an approximation of a variable or function. The aim of this sub-optimisation process is to obtain an approximate Pareto front for the problem. Following this, the set of solutions found are refined using AWS.

In step 2, the refined points found in step one are treated as feasible points for further sub-optimisation. An additional constraint H is added, which is a relaxation factor that allows for a larger search area around the feasible points in Step one, theoretically permitting the discovery of more solution points.

For a two-subsystem problem, subsystem i for this step of the optimisation is the same as that expressed in Equation B.8, with the addition of the constraint H_i

$$H_i(\mathbf{x}_i, \mathbf{y}_i, \tilde{\mathbf{y}}_j) \leq 0 \tag{B.9}$$

It is stated in the paper that the subsystem problem above can be extended to any number of subsystems, although the formulations given are specifically for a two-subsystem problem, and are therefore not generalised.

Appendix C

Coefficients for the static model

In this Appendix, the coefficients for the static models discussed in Chapter 7 are given. These coefficients are taken from [167].

C.1 Cluster 2

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.20928043 & 0 & 0.3273392 & -0.07295733 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.16411828 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.37573615 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2883655 & -0.45641958 & 0 & 0 & 0.156452 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.05755873 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.07966742 & 0 & 0 \end{bmatrix} \tag{C.1}$$

$$B = \begin{bmatrix} -0.0793759 & -0.193616 & 0.2857791 \\ 0 & 0 & 0 \\ 0.2723485 & 0 & 0 \\ -0.16861411 & 0 & 0 \\ 0.11069397 & 0 & 0 \\ 0.13378498 & -0.0304089 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{C.2}$$

$$C = \begin{bmatrix} 0.2673702 \\ -0.17359901 \\ -0.2138701 \\ 0.09668806 \\ 0.02593423 \\ 0.02295327 \\ -0.35942442 \\ 0.41436199 \end{bmatrix} \tag{C.3}$$

C.2 Cluster 3

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2005243 & 0 & 0.28223255 & 0.01133989 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.07900744 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.37816344 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.2430594 & -0.604806 & 0 & 0 & -0.01494081 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.01246921 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.04097719 & 0 & 0 \end{bmatrix} \tag{C.4}$$

$$B = \begin{bmatrix} 0.01638216 & -0.16459091 & 0.21911528 \\ 0 & 0 & 0 \\ 0.3584852 & 0 & 0 \\ -0.49364516 & 0 & 0 \\ -0.21091993 & 0 & 0 \\ 0.2255613 & -9.75356E-06 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{C.5}$$

$$C = \begin{bmatrix} -0.0897735 \\ 0.19763942 \\ 0.1286575 \\ -0.0834364 \\ -0.05920954 \\ 0.001659602 \\ 0.22164219 \\ -0.21716819 \end{bmatrix} \tag{C.6}$$

C.3 Cluster 4

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.28333974 & 0 & 0.37856233 & 0.08873795 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3572616 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.1453202 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.21101593 & -0.423738506 & 0 & 0 & 0.188294569 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.1301213 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.09234649 & 0 & 0 \end{bmatrix} \tag{C.7}$$

$$B = \begin{bmatrix} 0.0430726 & -0.2000715 & 0.4748308 \\ 0 & 0 & 0 \\ 0.1980498 & 0 & 0 \\ -0.2960459 & 0 & 0 \\ -0.0923919 & 0 & 0 \\ 0.188998916 & 0.038285893 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{C.8}$$

$$C = \begin{bmatrix} -0.5760012 \\ 0.50351111 \\ 0.7291977 \\ -0.2411204 \\ -0.4522848 \\ 0.001917455 \\ 0.9764244 \\ -1.10021169 \end{bmatrix} \tag{C.9}$$

Bibliography

[1] C. D. McAllister, T. W. Simpson, K. Hacker, K. Lewis, and A. Messac, "Integrating linear physical programming within collaborative optimization for multiobjective multidisciplinary design optimization," *Structural and Multidisciplinary Optimization*, vol. 29, pp. 178–189, Mar 2005.

[2] H. Li, M. Ma, and W. Zhang, "Multi-objective collaborative optimization using linear physical programming with dynamic weight," *Journal of Mechanical Science and Technology*, vol. 30, pp. 763–770, Feb 2016.

[3] A. Messac, "Physical programming - effective optimization for computational design," *AIAA Journal*, vol. 34, no. 1, pp. 149–158, 1996.

[4] C. McAllister, T. Simpson, K. Lewis, and A. Messac, "Robust multiobjective optimization through collaborative optimization and linear physical programming," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2012.

[5] H. Li, M. Ma, and W. Zhang, "Improving collaborative optimization for MDO problems with multi-objective subsystems," *Structural and Multidisciplinary Optimization*, vol. 49, pp. 609–620, Apr 2014.

[6] C. Baril, S. Yacout, and B. Clément, "An interactive multi-objective algorithm for decentralized decision making in product design," *Optimization and Engineering*, vol. 13, pp. 121–150, 2012.

[7] C.-H. Huang, J. Galuski, and C. L. Bloebaum, "Multi-objective Pareto concurrent subspace optimization for multidisciplinary design," *AIAA Journal*, vol. 45, no. 8, pp. 1894–1906, 2007.

[8] Y. Fan and B. Dino, “Genetic algorithm-based multiobjective optimization for building design,” *Architectural Engineering and Design Management*, vol. 6, no. 1, pp. 68–82, 2010.

[9] S. Rabeau, P. Dépincé, and F. Bennis, “Collaborative optimization of complex systems: a multidisciplinary approach,” *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 1, pp. 209–218, Nov 2007.

[10] V. Aute and S. Azarm, “A genetic algorithms based approach for multidisciplinary multiobjective collaborative optimization,” in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2006.

[11] S. Gunawan, S. Azarm, J. Wu, and A. Boyars, “Quality-assisted multi-objective multidisciplinary genetic algorithms,” *AIAA journal*, vol. 41, no. 9, pp. 1752–1762, 2003.

[12] M. Li and S. Azarm, “Multiobjective collaborative robust optimization with interval uncertainty and interdisciplinary uncertainty propagation,” *Journal of Mechanical Design*, vol. 130, Jul 2008.

[13] J. A. Duro, Y. Yan, R. C. Purshouse, and P. J. Fleming, “Collaborative multi-objective optimization for distributed design of complex products,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’18, (New York, NY, USA), p. 625–632, Association for Computing Machinery, 2018.

[14] A. Giassi, F. Bennis, and J.-J. Maisonneuve, “Multidisciplinary design optimisation and robust design approaches applied to concurrent design,” *Structural and Multidisciplinary Optimization*, vol. 28, pp. 356–371, Nov 2004.

[15] S. Parashar and C. Bloebaum, “Multi-objective genetic algorithm concurrent subspace optimization (MOGACSSO) for multidisciplinary design,” in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. 8, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, May 2006.

[16] Office for National Statistics, "Population estimates – local authority based by five year age band - Office for National Statistics," 2023. https://www.nomisweb.co.uk/query/construct/components/stdListComponent.asp?menuopt=12&subcomp=100, last accessed on 29/11/2023.

[17] Office for National Statistics, "Number and proportion of employee jobs with hourly pay below the living wage - Office for National Statistics," Oct 2022. https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/earningsandworkinghours/datasets/numberandproportionofemployeejobswithhourlypaybelowthelivingwage, last accessed on 25/10/2023.

[18] B. Brehmer, "Dynamic decision making: Human control of complex systems," *Acta Psychologica*, vol. 81, no. 3, pp. 211–241, 1992.

[19] H. Sillitto, R. Griego, E. Arnold, D. Dori, J. Martin, D. McKinney, P. Godfrey, D. Krob, and S. Jackson, "What do we mean by "system"? – system beliefs and worldviews in the incose community," *INCOSE International Symposium*, vol. 28, no. 1, pp. 1190–1206, 2018.

[20] D. Dori and H. Sillitto, "What is a system? an ontological framework," *Systems engineering*, vol. 20, no. 3, pp. 207–219, 2017.

[21] R. D. Arnold and J. P. Wade, "A definition of systems thinking: A systems approach," *Procedia Computer Science*, vol. 44, no. C, pp. 669–678, 2015.

[22] R. C. Purshouse, P. S. Meier, A. Brennan, K. B. Taylor, and R. Rafia, "Estimated effect of alcohol pricing policies on health and health economic outcomes in england: an epidemiological model," *The Lancet (British edition)*, vol. 375, no. 9723, pp. 1355–1364, 2010.

[23] L. Brennan, N. Sabounchi, A. Kemner, and P. Hovmand, "Systems thinking in 49 communities related to healthy eating, active living, and childhood obesity," *Journal of public health management and practice : JPHMP*, vol. 21 Suppl 3, pp. S55–69, 05 2015.

[24] J. Martins and A. Lambe, "Multidisciplinary design optimization: A survey of architectures," *AIAA Journal*, vol. 51, pp. 2049–2075, Sep 2013.

[25] L. A. Schmit, "Structural synthesis - its genesis and development," *AIAA Journal*, vol. 19, no. 10, pp. 1249–1263, 1981.

[26] J. Agte, O. de Weck, J. Sobieszczanski-Sobieski, P. Arendsen, A. Morris, and M. Spieck, "Mdo: assessment and direction for advancement—an opinion of one international group," *Structural and multidisciplinary optimization*, vol. 40, no. 1-6, pp. 17–33, 2010.

[27] H. Ashley, "On making things the best-aeronautical uses of optimization," *Journal of aircraft*, vol. 19, no. 1, pp. 5–28, 1982.

[28] F. Dexl, A. Hauffe, and K. Wolf, "Multidisciplinary multi-objective design optimization of an active morphing wing section," *Structural and Multidisciplinary Optimization*, vol. 62, pp. 2423–2440, Nov 2020.

[29] R. Ganguli and S. Rajagopal, "Multidisciplinary design optimization of an uav wing using Kriging based multi-objective genetic algorithm," in *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, vol. 36, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 1999.

[30] Y. Kim, Y.-H. Jeon, and D.-H. Lee, "Multi-objective and multidisciplinary design optimization of supersonic fighter wing," *Journal of Aircraft*, vol. 43, no. 3, pp. 817–824, 2006.

[31] P. M. Zadeh, M. Sayadi, and A. Kosari, "An efficient metamodel-based multi-objective multidisciplinary design optimization framework," *Applied Soft Computing*, vol. 74, pp. 760–782, 2019.

[32] S. Kodiyalam, "Multidisciplinary design optimization - some formal methods, framework requirements, and application to vehicle design," *International Journal of Vehicle Design - INT J VEH DES*, vol. 25, 01 2001.

[33] M. R. Kianifar and F. Campean, "Global optimisation of car front-end ge-

ometry to minimise pedestrian head injury levels," in *Proceedings of the Design Society*, vol. 1, (Cambridge), pp. 2873–2882, Cambridge University Press, 2019.

[34] N. Vlahopoulos and C. G. Hart, "A Multidisciplinary Design Optimization Approach to Relating Affordability and Performance in a Conceptual Submarine Design," *Journal of Ship Production and Design*, vol. 26, pp. 273–289, 11 2010.

[35] S. Wang, M. Yang, W. Niu, Y. Wang, S. Yang, L. Zhang, and J. Deng, "Multidisciplinary design optimization of underwater glider for improving endurance," *Structural and multidisciplinary optimization*, vol. 63, no. 6, pp. 2835–2851, 2021.

[36] S. E. Hannapel, *Development of multidisciplinary design optimization algorithms for ship design under uncertainty.* PhD thesis, University of Michigan, 2012.

[37] E. Campana, G. Fasano, and D. Peri, "Penalty function approaches for ship multidisciplinary design optimisation (mdo)," *European J. of Industrial Engineering*, vol. 6, pp. 765 – 784, 01 2012.

[38] E. F. Campana, G. Fasano, D. Peri, and A. Pinto, "Nonlinear programming approaches in the multidisciplinary design optimization of a sailing yacht keel fin," *Proceedings in applied mathematics and mechanics*, vol. 7, no. 1, pp. 1060601–1060602, 2007.

[39] H. Chen, W. Li, W. Cui, P. Yang, and L. Chen, "Multi-objective multidisciplinary design optimization of a robotic fish system," *Journal of Marine Science and Engineering*, p. 478, Apr 2021.

[40] W. Luo and W. Lyu, "An application of multidisciplinary design optimization to the hydrodynamic performances of underwater robots," *Ocean Engineering*, vol. 104, pp. 686–697, 2015.

[41] K. Klamroth, S. Mostaghim, B. Naujoks, S. Poles, R. Purshouse, G. Rudolph, S. Ruzika, S. Sayın, M. M. Wiecek, and X. Yao, "Multiobjective optimization

for interwoven systems," *Journal of multi-criteria decision analysis*, vol. 24, no. 1-2, pp. 71–81, 2017.

[42] P. Meier, R. Purshouse, M. Bain, C. Bambra, R. Bentall, M. Birkin, J. Brazier, A. Brennan, M. Bryan, J. Cox, *et al.*, "The SIPHER Consortium: Introducing the new UK hub for systems science in public health and health economic research," *Wellcome Open Research*, vol. 4, no. 174, 2019.

[43] H. R. Fazeley, H. Taei, H. Naseh, and M. Mirshams, "A multi-objective, multidisciplinary design optimization methodology for the conceptual design of a spacecraft bi-propellant propulsion system," *Structural and Multidisciplinary Optimization*, vol. 53, pp. 145–160, Jan 2016.

[44] L. Brevault, M. Balesdent, A. Hebbal, and A. P. de Mirand, "Surrogate model-based multi-objective MDO approach for partially reusable launch vehicle design," *AIAA Scitech 2019 Forum*, 2019.

[45] D. Yang, M. Turrin, S. Sariyildiz, and Y. Sun, "Sports building envelope optimization using multi-objective multidisciplinary design optimization (M-MDO) techniques: Case of indoor sports building project in China," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, (New York City, NY, USA), pp. 2269–2278, May 2015.

[46] HM Government, "How government works," 2024. `https://www.gov.uk/government/how-government-works`, Last accessed on 15/03/2024.

[47] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[48] N. P. Tedford and J. R. R. A. Martins, "Benchmarking multidisciplinary design optimization algorithms," *Optimization and Engineering*, vol. 11, pp. 159–183, 2010.

[49] B. Roth and I. Kroo, "Enhanced collaborative optimization: Application to an analytic test problem and aircraft design," in *12th AIAA/ISSMO Multidis-*

ciplinary Analysis and Optimization Conference, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2008.

[50] A. R. Dovi, G. A. Wrenn, J.-F. M. Barthelemy, P. G. Coen, and L. E. Hall, "Multidisciplinary design integration methodology for a supersonic transport aircraft," *Journal of aircraft*, vol. 32, no. 2, pp. 290–296, 1995.

[51] A. Viviani, A. Aprovitola, L. Iuspa, and G. Pezzella, "Aeroshape design of reusable re-entry vehicles by multidisciplinary optimization and computational fluid dynamics," *Aerospace Science and Technology*, vol. 105, p. 106029, 2020.

[52] M. Manikandan and R. S. Pant, "Multidisciplinary design approach for solar-powered tri-lobed halesa," *Energy Conversion and Management*, vol. 245, p. 114616, 2021.

[53] S. Jun, Y.-H. Jeon, J. Rho, and D.-h. Lee, "Application of collaborative optimization using genetic algorithm and response surface method to an aircraft wing design," *Journal of Mechanical Science and Technology*, vol. 20, pp. 133–146, Jan 2006.

[54] T. Kumano, S. Jeong, S. Obayashi, Y. Ito, K. Hatanaka, and H. Morino, "Multidisciplinary design optimization of wing shape for a small jet aircraft using Kriging model," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2006.

[55] Y. Guang-qiu, L. Shu-wen, and D. B. Perlurst, "Multidisciplinary design and collaborative optimization for excavator backhoe device," *Journal of Engineering Science and Technology Review*, vol. 7, no. 2, pp. 99–105, 2014.

[56] Y. Congyang, Z. Dequan, W. Chaoxian, Z. Lin, C. Tingting, J. Tien-Chien, and L. Juan, "Optimizing electric adjustment mechanism using the combination of multi-body dynamics and control," *Procedia Manufacturing*, vol. 35, pp. 1363–1369, 2019. The 2nd International Conference on Sustainable Materials Processing and Manufacturing, SMPM 2019, 8-10 March 2019, Sun City, South Africa.

[57] S. Zheng, J. Gao, and J. Xu, "Research on production planning and schedul-

ing based on improved collaborative optimization," *Concurrent Engineering*, vol. 27, pp. 99–111, Apr 2019.

[58] V. Johnson, J. A. Duro, V. Kadirkamanathan, and R. C. Purhouse, "A scalable test suite for bi-objective multidisciplinary optimisation," in *Proceedings of the 2023 International Conference Series on Evolutionary Multi-Criterion Optimization (EMO)*, (Cham), p. 319–332, Springer Nature Switzerland, Mar 2023.

[59] V. Johnson, J. A. Duro, V. Kadirkamanathan, and R. C. Purhouse, "Toward scalable benchmark problems for multi-objective multidisciplinary optimization," in *Proceedings of the 2022 IEEE Symposium Series On Computational Intelligence (IEEE SSCI)*, (New York City, NY, USA), pp. 133–140, Institute of Electrical and Electronics Engineers, Dec 2022.

[60] S. Padula, N. Alexandrov, and L. Green, "MDO test suite at NASA Langley Research Center," in *6th Symposium on Multidisciplinary Analysis and Optimization*, (Reston, VA, USA), pp. 410–420, Amer Inst Aeronautics Astronautics, 1996.

[61] H. Li, Y. Jing, S. Zhang, V. Ojleska, T. Kolemisevska, and G. Dimirovski, "Improved multidisciplinary multiobjective collaborative optimization technique based on NSGA-II algorithm," *IFAC Proceedings Volumes*, vol. 43, no. 25, pp. 139–144, 2010. 13th IFAC Workshop on Supplemental Ways for Improving International Stability.

[62] D. V. Steward, "The design structure system: A method for managing the design of complex systems," *IEEE Transactions on Engineering Management*, vol. EM-28, no. 3, pp. 71–74, 1981.

[63] A. B. Lambe and J. R. R. A. Martins, "Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes," *Structural and Multidisciplinary Optimization*, vol. 46, pp. 273–284, Aug 2012.

[64] K.-H. Chang, "Chapter 5 - Multiobjective optimization and advanced topics,"

in *Design Theory and Methods Using CAD/CAE* (K.-H. Chang, ed.), pp. 325–406, Boston: Academic Press, Jan 2015.

[65] J. S. Arora, "Introduction to optimization," in *Optimization of Structural and Mechanical Systems*, pp. 1–34, World Scientific, Sep 2007.

[66] R. Shi, T. Long, N. Ye, Y. Wu, Z. Wei, and Z. Liu, "Metamodel-based multidisciplinary design optimization methods for aerospace system," *Astrodynamics*, vol. 5, pp. 185–215, Sep 2021.

[67] T. C. Wagner and P. Y. Papalambros, "A general framework for decomposition analysis in optimal design," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 19th Design Automation Conference: Volume 2 — Design Optimization; Geometric Modeling and Tolerance Analysis; Mechanism Synthesis and Analysis; Decomposition and Design Optimization, (Hoboken, NJ, USA), pp. 315–325, John Wiley & Sons, Inc., Sep 1993.

[68] P. Dépincé, B. Guédas, and J. Picard, "Multidisciplinary and multiobjective optimization: Comparison of several methods," in *7th World Congress on Structural and Multidisciplinary Optimization*, (Seoul, Korea), May 2007.

[69] Z. Lu and J. R. R. A. Martins, "Graph partitioning-based coordination methods for large-scale multidisciplinary design optimization problems," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2012.

[70] S. Tosserams, A. T. Hofkamp, L. F. P. Etman, and J. E. Rooda, "A specification language for problem partitioning in decomposition-based design optimization," *Structural and Multidisciplinary Optimization*, vol. 42, pp. 707–723, Nov 2010.

[71] J. L. Rogers and C. Bloebaum, "Ordering design tasks based on coupling strengths," in *5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Reston, VA, USA), pp. 708 – 717, Amer

Inst Aeronautics Astronautics, 1994.

[72] N. F. Michelena and P. Y. Papalambros, “Optimal model-based decomposition of powertrain system design,” *Journal of Mechanical Design*, vol. 117, pp. 499–505, Dec 1995.

[73] N. F. Michelena and P. Y. Papalambros, “A hypergraph framework for optimal model-based decomposition of design problems,” *Computational Optimization and Applications*, vol. 8, pp. 173–196, Sep 1997.

[74] T. Dietz, K. Klamroth, K. Kraus, S. Ruzika, L. E. Schäfer, B. Schulze, M. Stiglmayr, and M. M. Wiecek, “Introducing multiobjective complex systems,” *European Journal of Operational Research*, vol. 280, no. 2, pp. 581–596, 2020.

[75] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala, “Organizing the tasks in complex design projects,” in *Computer-Aided Cooperative Product Development* (D. Sriram, R. Logcher, and S. Fukuda, eds.), (Berlin, Heidelberg), pp. 229–252, Springer Berlin Heidelberg, 1991.

[76] A. Kusiak and J. Wang, “Efficient organizing of design activities,” *International Journal of Production Research*, vol. 31, no. 4, pp. 753–769, 1993.

[77] A. Kusiak and J. Wang, “Decomposition of the design process,” *Journal of Mechanical Design*, vol. 115, pp. 687–695, Dec 1993.

[78] K. Lewis and F. Mistree, “Modeling interactions in multidisciplinary design: A game theoretic approach,” *Journal of Aircraft*, vol. 35, pp. 1387–1392, Aug 1997.

[79] M. Xiao, X. Shao, L. Gao, and Z. Luo, “A new methodology for multi-objective multidisciplinary design optimization problems based on game theory,” *Expert Systems with Applications*, vol. 42, no. 3, pp. 1602–1612, 2015.

[80] R. Sellar, S. Batill, and J. Renaud, “Response surface based, concurrent subspace optimization for multidisciplinary system design,” in *34th Aerospace Sciences Meeting and Exhibit*, (Reston, VA, USA), Amer Inst Aeronautics

Astronautics, 1996.

[81] T. Simpson, T. Mauery, J. Korte, and F. Mistree, "Comparison of response surface and Kriging models for multidisciplinary design optimization," in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, vol. 98, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, Sep 1998.

[82] X. Liu, Q. Yuan, M. Zhao, W. Cui, and T. Ge, "Multiple objective multidisciplinary design optimization of heavier-than-water underwater vehicle using cfd and approximation model," *Journal of Marine Science and Technology*, vol. 22, pp. 135–148, Mar 2017.

[83] W. Hu, S. Azarm, and A. Almansoori, "New approximation assisted multi-objective collaborative robust optimization (new AA-McRO) under interval uncertainty," *Structural and Multidisciplinary Optimization*, vol. 47, pp. 19–35, Jan 2013.

[84] E. Ghotbi and A. K. Dhingra, "A bilevel game theoretic approach to optimum design of flywheels," *Engineering Optimization*, vol. 44, no. 11, pp. 1337–1350, 2012.

[85] S. Bates, J. Sienz, and V. Toropov, "Formulation of the optimal Latin hypercube design of experiments using a permutation genetic algorithm," in *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2004.

[86] J. Maginot, *Sensitivity analysis for multidisciplinary design optmization*. PhD thesis, Cranfield University, UK, 2007.

[87] J. Sobieszczanski-Sobieski, "Optimization by decomposition: A step from hierarchic to non-hierarchic systems," Apr 1989.

[88] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization* (A. Abraham, L. Jain, and R. Goldberg, eds.), ch. Evolution-

ary Multiobjective Optimization: Theoretical Advances and Applications, pp. 105–145, Springer London, 2005.

[89] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[90] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Evolutionary Multi-Criterion Optimization* (C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, eds.), (Berlin, Heidelberg), pp. 280–295, Springer Berlin Heidelberg, 2005.

[91] D. Brockhoff, A. Auger, N. Hansen, and T. Tušar, "Using well-understood single-objective functions in multiobjective black-box optimization test suites," *Evolutionary Computation*, vol. 30, pp. 165–193, Jun 2022.

[92] T. Tušar, D. Brockhoff, N. Hansen, and A. Auger, "COCO: The Bi-objective Black Box Optimization Benchmarking (BBOB-BiObj) test suite." arXiv, 2019.

[93] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *Evolutionary Multi-Criterion Optimization* (E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, eds.), (Berlin, Heidelberg), pp. 284–298, Springer Berlin Heidelberg, 2001.

[94] S. Kodiyalam, "Evaluation of methods for multidisciplinary design optimization (MDO), Phase I," tech. rep., National Aeronautics and Space Administration, 1998.

[95] T. Ray, "Golinski's speed reducer problem revisited," *AIAA Journal*, vol. 41, no. 3, pp. 556–558, 2003.

[96] J. Golinski, "Optimal synthesis problems solved by means of nonlinear programming and random methods," *Journal of Mechanisms*, vol. 5, no. 3, pp. 287–309, 1970.

[97] D. Pate, J. Gray, and B. German, "A graph theoretic approach to problem

formulation for multidisciplinary design analysis and optimization," *Structural and Multidisciplinary Optimization*, vol. 49, pp. 743–760, May 2013.

[98] S. Delbecq, M. Budinger, and A. Reysset, "Benchmarking of monolithic MDO formulations and derivative computation techniques using OpenMDAO," *Structural and Multidisciplinary Optimization*, vol. 62, pp. 645–666, Aug 2020.

[99] J. Gray, K. T. Moore, T. A. Hearn, and B. A. Naylor, "Standard platform for benchmarking multidisciplinary design analysis and optimization architectures," *AIAA Journal*, vol. 51, no. 10, pp. 2380–2394, 2013.

[100] A. Kurapati and S. Azarm, "Immune network simulation with multiobjective genetic algorithms for multidisciplinary design optimization," *Engineering optimization*, vol. 33, no. 2, pp. 245–260, 2000.

[101] M. Farnsworth, A. Tiwari, M. Zhu, and E. Benkhelifa, "A multi-objective and multidisciplinary optimisation algorithm for microelectromechanical systems," in *NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop NEO 2016 and the NEO Cities 2016 Workshop held on September 20-24, 2016 in Tlalnepantla, Mexico* (Y. Maldonado, L. Trujillo, O. Schütze, A. Riccardi, and M. Vasile, eds.), pp. 205–238, Cham: Springer International Publishing, 2018.

[102] S. Gunawan, A. Farhang-Mehr, and S. Azarm, "On maximizing solution diversity in a multiobjective multidisciplinary genetic algorithm for design optimization," *Mechanics based design of structures and machines*, vol. 32, no. 4, pp. 491–514, 2004.

[103] D. Du and D. Simon, "Complex system optimization using biogeography-based optimization," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–17, Jan 2013.

[104] S. Gunawan, A. Farhang-Mehr, and S. Azarm, "Multi-level multi-objective genetic algorithm using entropy to preserve diversity," in *Evolutionary Multi-Criterion Optimization* (C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele,

and K. Deb, eds.), (Berlin, Heidelberg), pp. 148–161, Springer Berlin Heidelberg, 2003.

[105] M. Gardenghi and M. M. Wiecek, "Efficiency for multiobjective multidisciplinary optimization problems with quasiseparable subproblems," *Optimization and Engineering*, vol. 13, pp. 293–318, Jun 2012.

[106] M. R. Bonyadi, Z. Michalewicz, and L. Barone, "The travelling thief problem: The first step in the transition from theoretical problems to realistic problems," in *2013 IEEE Congress on Evolutionary Computation*, (New York City, NY, USA), pp. 1037–1044, Institute of Electrical and Electronics Engineers, 2013.

[107] G. B. Mathews, "On the partition of numbers," *Proceedings of the London Mathematical Society*, vol. s1-28, no. 1, pp. 486–490, 1896.

[108] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, "The traveling salesman problem: A guided tour of combinatorial optimization," *The Journal of the Operational Research Society*, vol. 37, no. 5, p. 535, 1986.

[109] M.-K. Shin and G.-J. Park, "Multidisciplinary design optimization based on independent subspaces," *International Journal for Numerical Methods in Engineering*, vol. 64, no. 5, pp. 599–617, 2005.

[110] F. Campelo and C. Aranha, "EC Bestiary: A bestiary of evolutionary, swarm and other metaphor-based algorithms," Jun 2018.

[111] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[112] S. Tosserams, L. F. P. Etman, P. Y. Papalambros, and J. E. Rooda, "An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers," *Structural and Multidisciplinary Optimization*, vol. 31, pp. 176–189, Mar 2006.

[113] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[114] Y. Y. Haimes and D. Li, "Hierarchical multiobjective analysis for large-scale systems: Review and current status," *Automatica*, vol. 24, no. 1, pp. 53–69, 1988.

[115] K. Miettinen, *Nonlinear Multiobjective Optimization [electronic resource]*. International Series in Operations Research & Management Science, 12, New York, NY: Springer US : Imprint: Springer, 1st ed. 1998. ed., 1998.

[116] J. Koski, "Defectiveness of weighting method in multicriterion optimization of structures," *Communications in Applied Numerical Methods*, vol. 1, no. 6, pp. 333–337, 1985.

[117] T. W. Athan and P. Y. Papalambros, "A note on weighted criteria methods for compromise solutions in multi-objective optimization," *Engineering Optimization*, vol. 27, no. 2, pp. 155–176, 1996.

[118] J. Allison, D. Walsh, M. Kokkolaras, P. Papalambros, and M. Cartmell, "Analytical target cascading in aircraft design," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2006.

[119] H.-y. Huang and D.-y. Wang, "Improvement of collaborative optimization," *Journal of Shanghai Jiaotong University (Science)*, vol. 15, pp. 172–177, Apr 2010.

[120] R. V. Tappeta and J. E. Renaud, "Multiobjective collaborative optimization," *Journal of Mechanical Design*, vol. Volume 2: 23rd Design Automation Conference, pp. 403–411, Sep 1997. V002T29A017.

[121] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski, "Multidisciplinary optimization methods for aircraft preliminary design," in *5th Symposium on Multidisciplinary Analysis and Optimization*, (Reston, VA, USA), pp. 697–707, Amer Inst Aeronautics Astronautics, 1994.

[122] R. J. Balling and C. A. Wilkinson, "Execution of multidisciplinary design optimization approaches on common test problems," *AIAA Journal*, vol. 35, no. 1, pp. 178–186, 1997.

[123] F. Yang, *Collaborative optimisation in building design with a Pareto-based genetic algorithm.* PhD thesis, Loughborough University, Sep 2018.

[124] M. A. Ilgin and S. Gupta, "Physical programming: A review of the state of the art," *Studies in Informatics and Control*, vol. 21, pp. 349–366, Dec 2012.

[125] C. D. McAllister and T. W. Simpson, "Multidisciplinary robust design optimization of an internal combustion engine," *Journal of mechanical design (1990)*, vol. 125, no. 1, pp. 124–130, 2003.

[126] G. Hernandez, J. Allen, and F. Mistree, "The compromise decision support problem: Modeling the deviation function as in physical programming," *Engineering Optimization*, vol. 33, pp. 445–471, Apr 2001.

[127] F. Mistree, O. Hughes, and B. Bras, *Compromise Decision Support Problem And The Adaptive Linear Programming Algorithm*, pp. 251–290. Reston, VA, USA: Amer Inst Aeronautics Astronautics, 1993.

[128] H.-Z. Huang, Y. Tao, and Y. Liu, "Multidisciplinary collaborative optimization using fuzzy satisfaction degree and fuzzy sufficiency degree model," *Soft Computing*, vol. 12, pp. 995–1005, Aug 2008.

[129] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—i," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[130] H.-Z. Huang, "Reliability evaluation of a hydraulic truck crane using field data with fuzziness," *Microelectronics Reliability*, vol. 36, no. 10, pp. 1531–1536, 1996.

[131] V. Johnson, J. A. Duro, V. Kadirkamanathan, and R. C. Purshouse, "A distributed multi-disciplinary design optimization benchmark test suite with constraints and multiple conflicting objectives," in *Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion*, GECCO '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[132] A. Farhang-Mehr and S. Azarm, "Diversity assessment of Pareto optimal solution sets: an entropy approach," in *Proceedings of the 2002 Congress on*

Evolutionary Computation. CEC'02 (Cat. No.02TH8600), vol. 1, (New York City, NY, USA), pp. 723–728 vol.1, Institute of Electrical and Electronics Engineers, 2002.

[133] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Parallel Problem Solving from Nature — PPSN IV* (H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, eds.), (Berlin, Heidelberg), pp. 584–593, Springer Berlin Heidelberg, 1996.

[134] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms.," in *Proceedings of the First Int. Conference on Genetic Algortihms, Ed. G.J.E Grefensette, J.J. Lawrence Erlbraum*, pp. 93–100, L. Erlbaum Associates Inc., Jan 1985.

[135] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, p. 221–248, Sep 1994.

[136] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[137] S. Narayanan and S. Azarm, "On improving multiobjective genetic algorithms for design optimization," *Structural optimization*, vol. 18, pp. 146–155, Oct 1999.

[138] W. Hu, S. Azarm, and A. Almansoori, "Approximation assisted multiobjective collaborative robust optimization (AA-McRO) under interval uncertainty," in *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2010.

[139] W. Hu, M. Li, S. Azarm, and A. Almansoori, "Multi-objective robust optimization under interval uncertainty using online approximation and constraint cuts," *Journal of Mechanical Design*, vol. 133, Jun 2011.

[140] M. Li, S. Azarm, and A. Boyars, "A new deterministic approach using sensitivity region measures for multi-objective robust and feasibility robust design

optimization," *Journal of mechanical design (1990)*, vol. 128, no. 4, pp. 874–883, 2006.

[141] M. Li, *Robust optimization and sensitivity analysis with multi-objective genetic algorithms: Single-and multi-disciplinary applications.* PhD thesis, University of Maryland, College Park, 2007.

[142] M. Li, J. Hamel, and S. Azarm, "Optimal uncertainty reduction for multi-disciplinary multi-output systems using sensitivity analysis," *Structural and Multidisciplinary Optimization*, vol. 40, pp. 77–96, Jan 2010.

[143] M. Li, N. Williams, and S. Azarm, "Interval uncertainty reduction and single-disciplinary sensitivity analysis with multi-objective optimization," *Journal of Mechanical Design*, vol. 131, Feb 2009.

[144] T. Xia and M. Li, "A new sequential robust approach for MDO problems under mixed interval and probabilistic uncertainties," in *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. Volume 2B: 44th Design Automation Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (Hoboken, NJ, USA), John Wiley & Sons, Inc., Aug 2018.

[145] T. Xia, M. Li, and J. Zhou, "A sequential robust optimization approach for multidisciplinary design optimization with uncertainty," *Journal of mechanical design (1990)*, vol. 138, no. 11, 2016.

[146] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part i: A unified formulation," *IEEE transactions on systems, man and cybernetics. Part A, Systems and humans*, vol. 28, no. 1, pp. 26–37, 1998.

[147] R. E. Steuer, *Multiple criteria optimization: theory, computation, and application.* Wiley series in probability and mathematical statistics. Applied probability and mathematical statistics, New York ; Chichester: Wiley, 1986.

[148] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive

multiobjective optimization: A review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41256–41279, 2018.

[149] B. Roth, *Aircraft family design using enhanced collaborative optimization*. PhD thesis, Stanford University, Jan 2008.

[150] I. Chittick and J. Martins, "An asymmetric suboptimization approach to aerostructural optimization," *Optimization and Engineering*, vol. 10, pp. 133–152, Mar 2009.

[151] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. R. A. Martins, "PyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems," *The Journal of Open Source Software*, vol. 5, no. 54, p. 2564, 2020.

[152] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, vol. 59, pp. 1075–1104, 2019.

[153] C. L. Bloebaum, "Coupling strength-based system reduction for complex engineering design," *Structural Optimization*, vol. 10, pp. 113—-121, 1995.

[154] C. M. Fonseca, L. Paquete, and M. López-Ibáñez, "An improved dimension–sweep algorithm for the hypervolume indicator," in *Proceedings of the 2006 Congress on Evolutionary Computation (CEC'06)*, pp. 1157–1163, IEEE Press, 2006.

[155] M. López-Ibáñez, L. Paquete, and T. Stützle, "Exploratory analysis of stochastic local search algorithms in biobjective optimization," in *Experimental Methods for the Analysis of Optimization Algorithms* (T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, eds.), pp. 209–222, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[156] R. D. Braun, P. Gage, I. M. Kroo, and I. Sobiesiki, "Implementation and performance issues in collaborative optimization," in *6th AIAA/NASA/ISSMO*

Symposium on Multidisciplinary Analysis and Optimization, pp. 295–305, 1996.

[157] N. Alexandrov and R. Lewis, "Comparative properties of collaborative optimization and other approaches to MDO," tech. rep., Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, USA, Mar 2000.

[158] J. Blank and K. Deb, "PyMoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[159] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[160] N. M. Alexandrov and R. M. Lewis, "Analytical and computational aspects of collaborative optimization for multidisciplinary design," *AIAA Journal*, vol. 40, no. 2, pp. 301–309, 2002.

[161] T. Cormier, A. Scott, L. Ledsinger, D. McCormick, D. Way, and J. Olds, "Comparison of collaborative optimization to conventional design techniques for a conceptual RLV," in *8th Symposium on Multidisciplinary Analysis and Optimization*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, Aug 2012.

[162] M. Hassannezhad, M. Gogarty, C. H. O'Connor, J. Cox, P. S. Meier, and R. C. Purshouse, "A cybernetic participatory approach for policy system of systems mapping: Case study of inclusive economies," *Futures*, vol. 152, p. 103200, 2023.

[163] Oxford Consultants for Social Inclusion (OCSI), "LSOAS,

LEPs and lookups: A beginner's guide to statistical geographies," Apr 2023. ocsi.uk/2019/03/18/lsoas-leps-and-lookups-a-beginners-guide-to-statistical-geographies, last accessed on 25/10/2023.

[164] Office for National Statistics, "Census 2021 geographies - Office for National Statistics," Dec 2021. www.ons.gov.uk/methodology/geography/ukgeographies/censusgeographies/census2021geographies, last accessed on 25/10/2023.

[165] S. Noble, D. McLennan, M. Noble, E. Plunkett, N. Gutacker, M. Silk, and G. Wright, "The English Indices of Deprivation 2019," research report, Ministry of Housing, Communities and Local Government, Sep 2019.

[166] P. Li, D. Veres, and V. Kadirkamanathan, "State space dynamic systems modelling of the inclusive economy and health outcomes," 2023. SIPHER Project WS4 Technical Report.

[167] D. Veres, P. Li, and V. Kadirkamanathan, "A data driven static systems map for the gmca inclusive economy," 2024. Unpublished Technical Note.

[168] K. Miettinen, F. Ruiz, and A. P. Wierzbicki, *Introduction to Multiobjective Optimization: Interactive Approaches*, pp. 27–57. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[169] J. Allison, M. Kokkolaras, M. Zawislak, and P. Papalambros, "On the use of analytical target cascading and collaborative optimization for complex system design," in *6th World Conference on Structural and Multidisciplinary Optimization*, (Rio de Janeiro), May 2005.

[170] W. Ting-ting, C. Xiao-kai, and L. Yi, "Multi-objective optimization based on the integration of linear physical programming within analytical target cascading," in *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, vol. 4, (New York City, NY, USA), pp. 2286–2289, Institute of Electrical and Electronics Engineers, 2011.

[171] N. Kang, M. Kokkolaras, and P. Y. Papalambros, "Solving multiobjective

optimization problems using quasi-separable MDO formulations and analytical target cascading," *Structural and Multidisciplinary Optimization*, vol. 50, pp. 849–859, Nov 2014.

[172] M. Kalsi, K. Hacker, and K. Lewis, "A comprehensive robust design approach for decision trade-offs in complex systems design," *Journal of mechanical design (1990)*, vol. 123, no. 1, pp. 1–10, 2001.

[173] C.-H. Huang, *Development of Multi-Objective Concurrent Sub-space Optimization and Visualization Methods for Multidisciplinary Design*. PhD thesis, Buffalo, NY: State University of New York at Buffalo, 2003.

[174] C.-H. Huang and C. Bloebaum, "Visualization as a solution aid for multi-objective concurrent subspace optimization in a multidisciplinary design environment," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2004.

[175] C.-H. Huang and C. Bloebaum, "Incorporation of preferences in multi-objective concurrent subspace optimization for multidisciplinary design," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Reston, VA, USA), Amer Inst Aeronautics Astronautics, 2004.

[176] K. F. Hulme and C. L. Bloebaum, "Development of a multidisciplinary design optimization test simulator," *Structural optimization*, vol. 14, pp. 129–137, Oct 1997.

[177] K. Zhang, W. Li, and W. Song, "Bi-level adaptive weighted sum method for multidisciplinary multi-objective optimization," in *46th AIAA Aerospace Sciences Meeting and Exhibit*, (Reston, VA, USA), pp. 2611–2622, Amer Inst Aeronautics Astronautics, 2008.

[178] I. Y. Kim and O. L. de Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Structural and Multidisciplinary Optimization*, vol. 29, pp. 149–158, Feb 2005.

[179] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization prob-

lems," *Structural optimization*, vol. 14, pp. 63–69, Aug 1997.

[180] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Structural optimization*, vol. 10, pp. 94–99, Oct 1995.