How to select MDAO workflows

Sanchez Perez Moreno, Sebastian; Zaaijer, Michiel B

DOI
10.2514/6.2018-0654

Publication date
2018

Document Version
Final published version

Published in
2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference

AIAA SciTech Forum
8–12 January 2018, Kissimmee, Florida
2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference

10.2514/6.2018-0654

How to select MDAO workflows

Sebastian Sanchez Perez-Moreno* and Michiel B. Zaaijer†
Delft University of Technology, Delft, the Netherlands

Multidisciplinary Design Analysis and Optimisation (MDAO) workflows consist of coupled tools driven by an algorithm for a specific purpose, e.g. optimisation. MDAO users may have at their disposal a set of tools of varying levels of fidelity. As a result, many permutations or MDAO workflows may arise, for which no clear methodology exists to evaluate, compare and rank them based on their performance. Our research question is then how to find the most useful MDAO workflows for a given purpose. This paper provides a guideline for solving this multiple criteria decision analysis problem. Our guideline includes a method to define the criteria and metrics for evaluating the performance of MDAO workflows, a strategy to aggregate the scores, and an optimisation algorithm for categorical variables used to find the best alternatives. We apply this guideline to the offshore wind farm layout optimisation problem to demonstrate its use. This case study evidenced how critical the list of criteria is and that it should be built with qualitative and quantitative methods.

Nomenclature

C_k	Score of criterion k
R_j	Constraint on criterion j
W_i	MDAO analysis block i
Superscript	
i	Wind farm design i
Acronyms	
BEM	Blade Element Momentum theory
IEA	International Energy Agency
LCOE	Levelised Cost of Energy
MCDA	Multiple Criteria Decision Analysis
MDAO	Multidisciplinary Design Analysis and Optimisation
MOPSOC	Multiple Objective Particle Swarm Optimisation for Categorical variables
WFLOP	Wind Farm Layout Optimisation Problem
XDSM	Extended Design Structure Matrix

I. Background

Multidisciplinary Design Analysis and Optimisation (MDAO) stands for the practice of coupling multiple computational tools to a *driver* for solving a problem that requires the calculation of the overall performance of a system[1]. Each tool predicts the behaviour of a component of the system or its response as regards a specific physical discipline. The coupled tools (called from here onward *analysis block*) are then repeatedly called by a *driver*. We use the term *driver* in this context to refer to any algorithm that calls the analysis block for a specific purpose. We call this purpose a *use case*. Examples of drivers include optimisation algorithms, uncertainty quantification methods, sensitivity analyses and design certification with respect to multiple cases. Although MDAO originated for optimising systems with respect to design variables that describe multiple components, there is agreement that MDAO has evolved to include other drivers. A diagram of the simplest case of an MDAO workflow with two coupled modules is shown in Fig. 1.

*Ph.D. Researcher, Wind Energy section, Faculty of Aerospace Engineering, Member AIAA.
†Assistant Professor, Wind Energy section, Faculty of Aerospace Engineering.

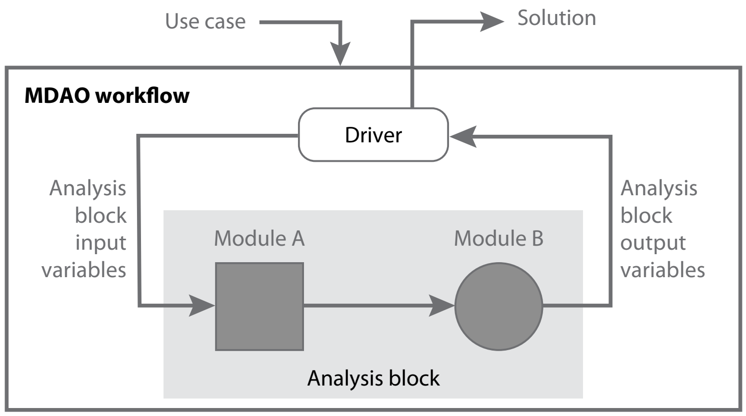

Figure 1. Diagram of an MDAO workflow. The analysis block, modules and driver are recognisable.

It is acknowledged that computational tools of varying levels of fidelity may be used to simulate and analyse the same component[1]. Since multiple tools are coupled in an MDAO workflow, there may be more than one possible combination, i.e. different MDAO workflows may exist for the same purpose.

At present, MDAO users usually provide qualitative reasons for the selection of tools and the driver, if at all. Typical arguments are that tools are selected for being the fastest, "highest fidelity", in-house built or the "only ones available".

We argue that while their choice can yield acceptable results, MDAO users are missing out on the possibility of improving the performance of their MDAO workflow by not exploring the coupling of other tools with different levels of fidelity. Likewise, the choice researchers make for the algorithm of a driver is often based on intuition, without testing its performance and comparing between alternatives[1].

II. The case for MDAO workflow evaluation

It is not surprising that the analysis block of a given MDAO workflow will perform better for some use cases than others.

The reader can understand it by picturing two nearly impossible extreme scenarios: the minimisation of the wake effects inside a wind farm with respect to its layout using the solution of the full Navier-Stokes equation to resolve the blade geometry; or analysing the 90-percentile of the cost of the energy produced by a wind plant to make an investment decision using a lookup table made with an empirical model. In both cases, the tools used have been poorly chosen since their best attributes are not fully exploited. In the first scenario, the optimisation would be prohibitively expensive and time consuming due to the consideration of irrelevant details. This modelling approach would result, however, in a highly detailed analysis block that could be used as a reference for the benchmark of simpler wake models. On the other hand, in the second scenario the uncertainty resulting from such an unsophisticated model would be unacceptably high for the purpose of financial investment and lead to unreliable decisions. Nevertheless. this modelling approach would be beneficial for an early design stage of the wind plant layout. In other words, the usefulness of the analysis block depends on its use case.

Similarly, drivers are favoured for analysis blocks with specific behaviour. Indeed, in the case of an optimisation driver, the shape and smoothness of the response surface can help inform the choice of algorithms. For example, designers agree that gradient-based optimisation algorithms perform better with smooth and continuous response surfaces, while their non-gradient based counterparts are better suited for functions with many local optima or discontinuous functions.

As a result, this work takes on the task of providing a guideline to evaluate, compare and rank a set of MDAO workflows by their usefulness for a given purpose or use case. The following guideline is envisioned to be:

- **concrete**, meaning that it is understandable and unambiguous;
- **flexible**, so that it can accommodate the specific requirements of MDAO users and adapt it to their own context;

- **objective**, in that the opinion of the user has minimum impact on the choice of tools to be used and thus providing the guideline with higher credibility;
- **simple**, for wide deployment, accessibility and acceptability.

The MDAO community benefits from this research as the ever-present trade-off between the sophistication and cost of multidisciplinary analysis and optimisation workflows continues to be overlooked.

III. Guideline for MDAO workflow selection

We have divided the guideline for selecting the most useful MDAO workflows in two phases: selecting first the most useful MDAO analysis blocks and then the most useful driver algorithms. Attempting to choose the MDAO workflow in its entirety at once is intractable due to the high computational burden of scoring the metrics that we will propose for the judgement of workflows. We therefore neglect any driver-analysis interactions. This is considered a reasonable assumption because the analysis block simulates a part of reality while the driver is a mathematical artificial technique that determines which realisations are evaluated. The performance of the driver therefore relates directly to the nature of reality and only indirectly to the nature of the analysis block by which reality is represented. Each phase in the selection is further broken down into three concepts: evaluation, comparison and ranking. We describe the guideline by referring to the diagram shown in Fig. 2. Phase 1 is a multicriteria decision problem that yields the most useful analysis blocks by means of a multiobjective optimisation, while phase 2—also a multicriteria decision problem—yields the most useful driver algorithms by means of applying MCDA techniques. Combined, they yield the most useful MDAO workflows.

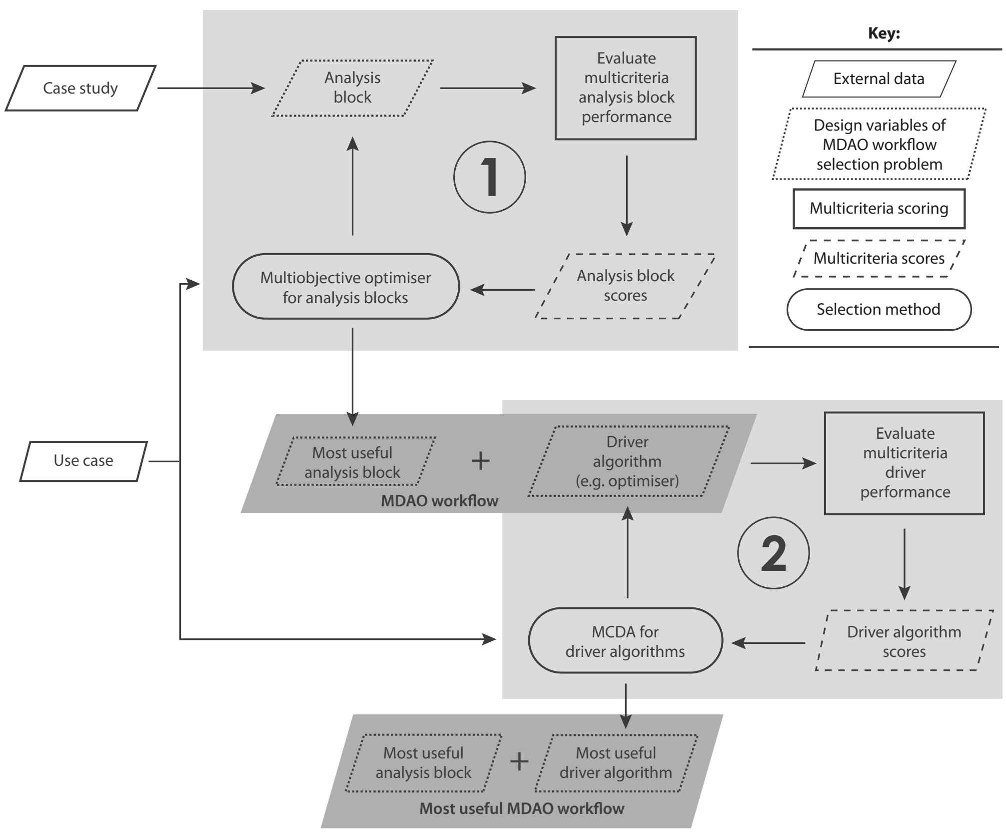

Figure 2. Diagram of the phases and procedures described in the guideline for MDAO workflow selection.

As identified by the IEA Wind Task 3 for Wind Energy Systems Engineering: Integrated RD&D[2], three dimensions of MDAO have to be explored in order to judge the workflow: model fidelity, system scope and MDAO architecture. However, in order to manage the high complexity of this problem, we have assumed a constant MDAO architecture and a fixed system scope, and thus only focus on the selection of model fidelity for every module. This means that the user has identified and chosen all the modules and their couplings (fixed the system scope), the type of driver to call the analysis block, as well as the input variables to the analysis block), the variables to be varied by the driver (e.g. design variables) and the output from the analysis needed by the driver.

In order to generate the alternatives that will be subject to evaluation, comparison and ranking, feasible MDAO workflows have to be instantiated with the permutations of the tools available. At this point, all unfeasible workflows (which cannot be connected due to input-output variables inconsistencies) are discarded. See Fig. 3 for visualising the following example: If we have tools $A1$ and $A2$ available for module A, and tools $B1$ and $B2$ for module B, the four possible analysis blocks that can result are $A1 - B1$, $A1 - B2$, $A2 - B1$, $A2 - B2$. Of these, we can discard the workflow containing $A1$ and $B2$, as $B2$ requires inputs that cannot be obtained from the outputs of $A1$. We would then have three feasible MDAO workflows from which to select the most useful (workflows 1, 2 and 4).

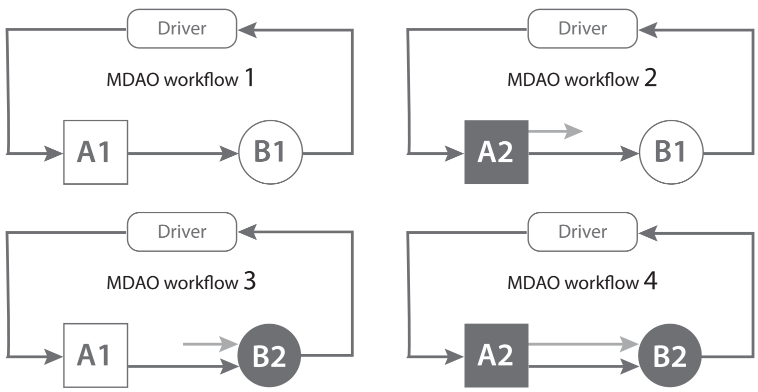

Figure 3. Diagram of four possible MDAO workflows where two modules can be filled by two tools each. MDAO workflow 3 is the only non-feasible alternative.

Having set the alternatives from which to select the best MDAO workflows, the user can define further requirements on the usefulness of the MDAO workflow (constraints to the selection problem), such as placing a limit on the execution time or memory use, or to keep only workflows with tools that have analytic derivatives.

III.A. Phase 1: selection of the analysis blocks

The selection of the most useful analysis blocks involves three activities:

1. **evaluation**: defining the criteria and metrics that specify what is meant by *useful* and with which an analysis block can be judged,
2. **comparison**: establishing the rules with which any two analysis blocks can be compared to each other,
3. **ranking**: optimising the module selection with respect to the comparison rules to identify the best performing analysis blocks.

The next sections describe these three activities.

III.A.1. Evaluation

Key in this work is to recognise that the behaviour of a system cannot be predicted by aggregating the behaviour of its components, but it will also be a function of their interactions. This remark leads us to suspect, likewise, that the performance of an MDAO analysis block cannot be estimated by measuring the performance of its isolated constituent modules, and instead we assess the performance of the analysis block as a whole.

The goal of an MDAO workflow is to capture the interactions between components and disciplines in a system of interest and exploit that additional knowledge during the design process. An example follows of a two-way interaction between phenomena in a wind farm that translates into a key interaction between the accuracy of two coupled modules.

The total energy harvested by a wind farm is a function of the local wind speed experienced by each wind turbine, which in turn is a function of the wake effects between turbines. In addition, the energy converted by every wind turbine is a function of their availability (the fraction of time that they are operational). Consequently, in order to calculate the total electrical energy converted by a wind farm more realistically, both the wind farm wake effects and the availability of the turbines—among others—should be taken into consideration. Let us imagine a researcher that has an analysis block that includes the simulation of both phenomena to estimate the energy converted by a wind plant. He also sets out to measure the electricity converted by a real wind farm for benchmarking the accuracy of the two analyses independently. He therefore collects a time series of the local wind speed experienced by every turbine, the time each one is operational and the total energy produced by the wind farm. The reader will agree that the total energy produced by the wind plant will naturally be determined by the wake effects and availabilities. However, it will also include two interactions: first, the wake behind every turbine is responsible for higher induced loads at the downstream wind turbines, which lead to higher failure rates and thus lower availability. Second, a non-operational wind turbine due to a failure or planned maintenance will lead to a change in the farm wake effects. Hence, there is no measured data to validate each tool separately (as we cannot decouple real phenomena affecting a system) and using the available data results in a rather spurious validation activity. Thus, our imaginary researcher should—as shall we—instead evaluate the accuracy of the coupled analysis block using the reference data.

Notwithstanding, we acknowledge that currently the only feasible way of validating modules that simulate individual system components is by isolating the component in an experimental set-up that avoids complex interactions, even if it is not representative of the real environment.

Since the usefulness of an MDAO workflow relies on several, conflicting criteria, we treat the selection of MDAO workflows as a Multiple Criteria Decision Analysis (MCDA) problem. MCDA helps a decision maker resolve the trade-offs between criteria[3].

Examples of objectives and concepts that may be of relevance to the evaluation of an MDAO analysis block include: accuracy, precision, repeatability, detail, range, resolution, sensitivity, CPU time, convergence, parallelism, feasibility, robustness, presence of analytical derivatives, availability, integrability, interoperability, causality, consistency, programming complexity, numerical stability, temporality, dependency, accountability, augmentability, communicativeness, completeness, conciseness, device-independence, efficiency, legibility, self-containedness, self-descriptiveness, structuredness and open-sourceness[4,5,6,7]. The exact interpretation of these criteria is still a matter of debate.

In order to shortlist the criteria used to evaluate the available MDAO workflows, we propose a top-down approach where we recursively answer the question: What makes an MDAO workflow [*(sub-)objective here*]? For example, we start with the question **Q**: What makes an MDAO workflow useful? **A**: its practicality and suitability to solve the problem; and then **Q**: What makes an MDAO workflow practical? **A**: if it achieves a solution fast and using available resources efficiently; and so on. In a bottom-up approach, in contrast, the meaningful differences between the alternatives are listed and then structured to higher level objectives. In this way a criteria tree is progressively built where nodes are sub-objectives and the branches at the lowest levels are the criteria to be measured. We provide an example of a criteria tree for judging MDAO analysis blocks in Fig. 4.

The difficult task is, however, to identify the relevant and most useful criteria for each specific use case and set of available alternatives. The reason why the set of alternatives may impact the criteria tree development process is that in practice, the alternatives will share certain attributes and differ in others. It is their differences what needs to be detected by the criteria. In any case, Keeney and Raiffa[8,9] state that the list of criteria with which MCDA shall be performed must comply with five attributes: completeness, operability, decomposability, non-redundancy and size. The meaning of these five attributes is explained below, along with a discussion of how they are addressed in our guideline.

Completeness addresses the adequacy of the list of criteria to meet the overall objective (in this case usefulness) and if its sub-objectives cover all areas of concern related to the performance of an MDAO workflow. The list of criteria is comprehensive if the decision maker gets an idea of the extent of achieving sub-objectives by measuring their corresponding criteria. A test for completeness entails logical deductive and inductive reasoning for proving that no gaps are left by the chosen objectives and criteria. The goal of this activity is to have the list fully describe the utility of each alternative. By making a criteria tree with our proposed methodology, gaps are more easily identified and dealt with from their early inception.

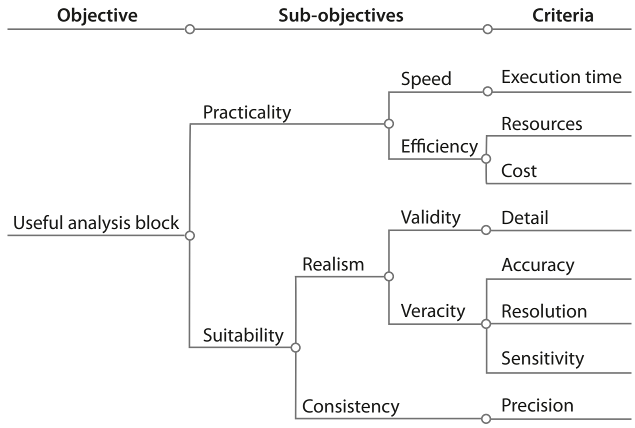

Figure 4. Example of a criteria tree for the multiple criteria decision analysis of MDAO analysis blocks.

Operability means that criteria must be meaningfully used in MCDA, have metrics that make concepts measurable, be understandable and pragmatic, and be useful for making decisions. Measurable criteria can be assigned a value or probability distribution and reflect the decision maker's preference between alternatives by being able to have different values. Criteria are responsible for advocating for a particular alternative, so they should represent differences in the alternatives. If all available alternatives have the same negative or positive aspect, then a criterion expressing that aspect will not comply with the operability attribute. We address operability in this guideline by means of argumentation. We establish the premise that the predefined metrics explain the variability in the performance of the MDAO workflows and deduct their operability from the scores of the alternatives throughout the MCDA.

Decomposability refers to the capacity of a list of criteria to be arranged in the form of a tree. A decomposable list of criteria allows the decision making problem to be disaggregated into smaller problems, since criteria will fit a hierarchy. Furthermore by decomposing a problem we guarantee that every criterion can be measured at a time and they should have no implications on other criteria. Our list building approach also guarantees decomposability.

Non-redundancy in the list of criteria strives to avoid double counting any effect. Criteria must be pairwise independent. There exist a number of correlation measures that determine the degree of independence between the scores of any two criteria, and a correlation matrix helps determine whether some criteria can be discarded or combined. We stress the fact that non-redundancy is tested for the list of criteria that govern the selection problem at hand, and does not attempt to elevate the correlation between criteria to an absolute truth.

Size refers to the number of criteria and should be kept as small as possible. There is great value in avoiding unnecessary complexity.

Every use case might require MDAO workflows to have different sub-objectives, e.g. an optimisation would have high optimality as one of its goals, whereas an uncertainty quantification use case would care for a high convergence rate based on sample size.

Furthermore, every criterion needs a monotonic metric that represents the quantitative desirability of an alternative with respect to that criterion. The requirement for monotony on the criteria metrics guarantees transitiveness between the decision maker's preferences of alternatives.

It is worth noting that some criteria additionally will need to have a referent defined, in order to indicate how well the MDAO analysis block or entire workflow represents it (e.g. an output value should match a reference value determined by another method, either a measurement or simulation), while other criteria will have as referent a desired value (e.g. execution time should approach zero).

III.A.2. Comparison

Once we have defined the criteria and metrics that evaluate the performance of an analysis block in the previous step, we can score the alternative analysis blocks on these individual criteria. However, as mentioned before, the criteria may be conflicting, and we have to aggregate the scores to be able to compare the blocks against our main objective: usefulness.

We interpret the concept of *solving* the trade-offs between criteria in this phase as finding the non-dominated solutions. Given two MDAO analysis blocks, W_1 and W_2, we use the standard definition of dominance: W_1 dominates W_2 if it satisfies the following two conditions:

$$\begin{aligned} &1. \quad C_k(W_1) \leq C_k(W_2) \quad \forall k \in (1, \ldots, n), \\ &2. \quad \exists k \in (1, \ldots, n) \ni C_k(W_1) < C_k(W_2) \end{aligned}$$

where n is the number of criteria and $C_k(W_l)$ is the score of the l-th MDAO analysis block with respect to the k-th criterion.

Due to the conflicting nature of the multiple criteria, usually no single alternative achieves the best score with respect to all criteria, and instead the set of non-dominated alternatives form the Pareto front.

III.A.3. Ranking

Having introduced the concept of non-dominance, we are now in place to provide the procedure by which we find the most preferred alternatives on the Pareto front.

While the Pareto front be can found by scoring and comparing all alternatives pairwise, this becomes unfeasible when the MDAO analysis block has several modules and there are several tools available for each module too. Our proposal is to approximate the Pareto front by means of a combinatorial optimisation algorithm. The multi-objective optimisation algorithm works by minimising all objective functions, so lower scores in metrics are expected to represent a higher preference for that alternative. In this optimisation the problem formulation is stated as:

$$\begin{aligned} &\underset{W}{\text{minimize}} && C_i(W) \quad i \in (1, \ldots, n), \\ &\text{subject to:} && C_j \leq R_j \quad j \in (1, \ldots, n) \end{aligned}$$

where we have assumed that two subsets: the criteria to be minimised (C_i), and criteria to be constrained (C_j) are subsets of the set of criteria. These need not be mutually exclusive. R stands for the criteria scores that the MDAO user may have set as constraints, and W is the categorical vector that defines the tools implemented in the analysis block.

The choice of which criteria to use as objective functions or constraints is for the MDAO user to make. The use case can provide significant hints towards this goal, e.g. an optimisation may benefit from having CPU time of the analysis block as one of the objective functions, and keeping its accuracy as a constraint, as the overall goal is not to reduce the error with respect to a referent, but to capture the trends in the behaviour of the system performance with respect to the design variables.

The design variables of the optimisation formulation at hand are categorical by nature (Tool A, Tool B, etc), while the objective functions are continuous (the scores of the analysis blocks on the metrics for the criteria). As a consequence, we aim to use a multiple objective optimiser for categorical variables.

Although a genetic algorithm for this type of problems exists,[10] we set out to develop a new one based on the particle swarm optimisation (PSO) algorithm[11]. PSO is a family of nature-inspired algorithms, where a swarm of particles traverses the design space, where every particle is influenced by a combination of its individual cognition and the collective behaviour of the swarm[12]. The rationale is that PSO algorithms converge faster than genetic based algorithms as the latter rely on long-term evolution while the former aggregates the short-term knowledge of the swarm,[13] and their exploration capabilities for finding the global optimum can be matched by adding a turbulence—sometimes referred to as craziness—variable.

The new Multiple Objective Particle Swarm Optimisation algorithm for Categorical Variables (MOPSOC) uses probability distribution functions as design variables instead, and the Pareto front (even if non-convex) is approximated by using dynamic weight aggregation and an archive of non-dominated solutions. For more information and validation see reference[11].

The output of the MOPSOC algorithm is an approximation of the set of analysis blocks that dominate all others, across multiple criteria. As explained before, this is the approximated Pareto front. These analysis blocks are candidates to be included in the most useful and thus optimal MDAO workflows.

III.B. Phase 2: selection of the driver algorithm

The set of analysis blocks deemed the most useful for the predefined use case in phase 1 are now coupled to a driver at the top level, as per the predefined architecture for the MDAO workflows. In phase 2, thus, entire MDAO workflows are selected (see Fig. 2), where the alternatives result from permuting the available driver algorithms and the set of analysis blocks found in phase 1.

As opposed to phase 1, there will only be a few alternative driver algorithms to select from in phase 2. The consequence is that the selection process can be expected to differ.

The guideline for phase 2 is divided in the same three aspects as phase 1: evaluation, comparison and ranking (explained in (III.A). We elaborate on the methods and definitions used in each aspect.

III.B.1. Evaluation

The process for evaluating MDAO workflows continues by establishing the criteria and metrics with which the performance of a driver is assessed. We have covered this process in §III.A.1 for the analysis block, and it applies identically for making a criteria tree for evaluating a driver algorithm. We provide an example of a criteria tree for evaluating MDAO driver algorithms in Fig. 5.

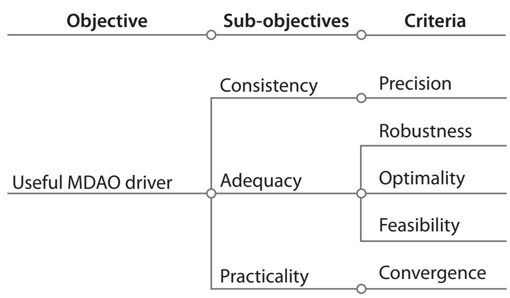

Figure 5. Example of a criteria tree for the multiple criteria decision analysis of MDAO driver algorithms.

Examples of criteria for the evaluation of MDAO drivers include precision, repeatability, convergence, sensitivity, CPU time, optimality, parallelism, feasibility, robustness, integrability, programming complexity, numerical stability, efficiency, legibility and open-sourceness.

III.B.2. Comparison

The concept of non-dominance introduced in §III.A.2 applies to phase 2 as well.

One *a posteriori* method to compare the alternatives within the Pareto front, is to use their distance to the utopia point[14]. The non-existing alternative that would have the best scores per criterion found across the entire space of feasible alternatives is called the utopia point. The smaller the Euclidean distance between an alternative and the utopia point, the better that alternative is considered. This approach is analogous to reducing the loss incurred at any given criterion.

III.B.3. Ranking

Provided that the number of alternatives in phase 2 is expected to be in the order of tens, the entire set of alternatives may be evaluated without incurring in extreme costs or use of resources. Therefore, we suggest scoring all alternatives for all criteria and using the ϵ-non-dominated sort algorithm[15]. This algorithm ranks all alternatives and provides the Pareto front of a given set.

Having found the set of non-dominated alternatives, we further propose to rank them by their distance to the utopia point.

IV. Case study: Offshore wind farm layout optimisation

IV.A. Use case and problem formulation

Due to the complexity of wind energy systems and the large number of disciplines involved in their design, practitioners have recently started applying MDAO in this domain[1]. We have applied the proposed guideline to select the optimal MDAO workflow for an offshore wind farm layout optimisation problem (WFLOP). The objective is to find the optimal placement of wind turbines such that the levelised cost of energy (LCOE) of an offshore wind plant is minimised. The design variables are the Cartesian coordinates of the wind turbines, and constraints include the boundaries of the area assigned and a minimum distance between the wind turbines to avoid collisions. This is expressed as:

$$
\begin{array}{ll}
\underset{\mathbf{x_i} \quad i \in (1,\ldots,N)}{\text{minimize}} & LCOE(x_i) \\
\text{subject to:} & distance(\mathbf{x_i}, \mathbf{x_j}) > D \quad \forall i \neq j \quad i,j \in (1,\ldots,N) \\
 & \mathbf{x_i} \in S \quad \forall i \in (1,\ldots,N),
\end{array}
$$

where x_i are the Cartesian coordinates of turbine i, D is the turbine diameter and S is the assigned area.

The scenario used for this case is a hypothetical square region in the North Sea where nine 5 MW NREL wind turbines[16] are to be installed.

We assume an early stage design where we seek to find feasible layouts whose LCOE responds to the interactions between disciplines. An extended design structure matrix (XDSM)[17] of all MDAO workflows considered is shown in Fig. 6.

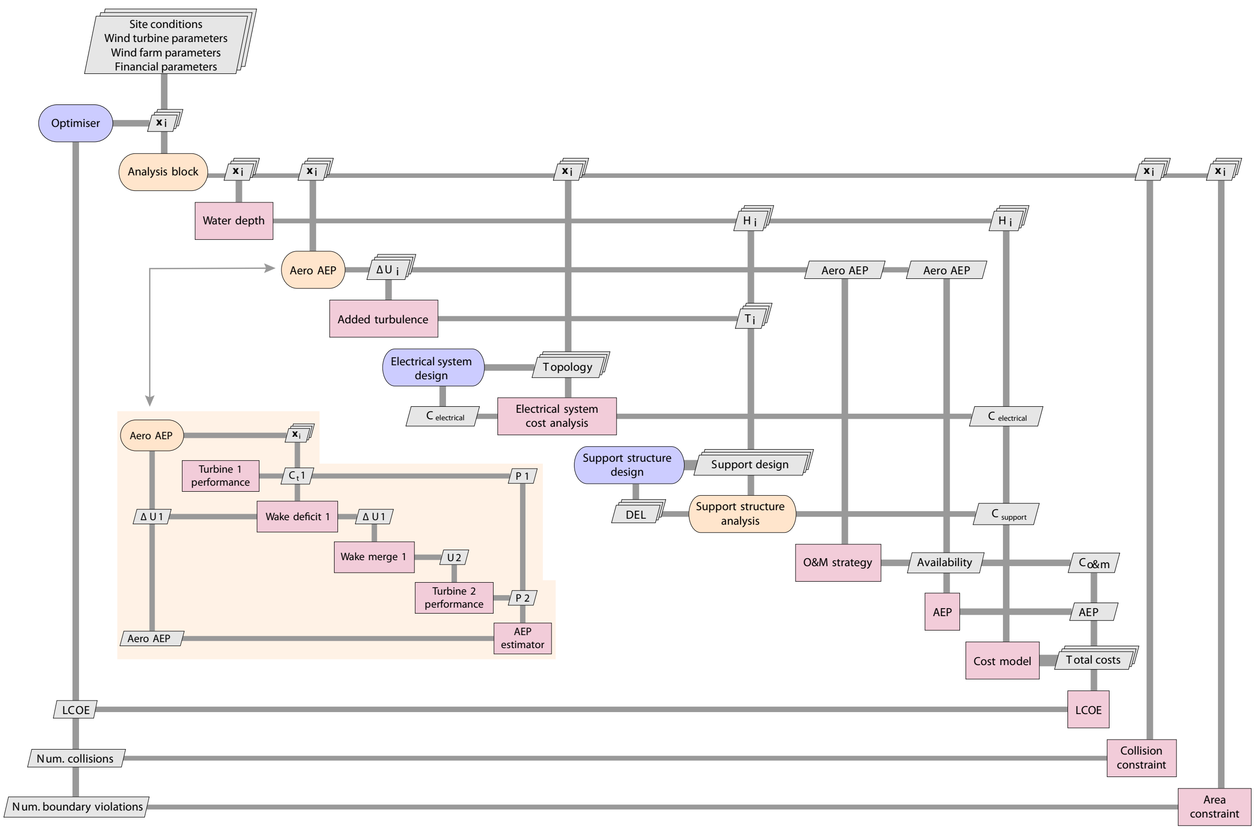

Figure 6. Extended design structure matrix of the MDAO workflows for an offshore wind farm layout optimisation problem.

IV.B. List of alternatives: feasible MDAO workflows

We outline here the alternatives from which we want to select our MDAO workflow.

The MDAO analysis blocks are built by permuting the set of tools presented in Table 1. Certain modules have only one tool available and are thus not listed, as that tool is included in all alternatives. In addition, we consider trivial constant output tools for some modules, with the purpose of testing this guideline.

Moreover, the optimisation algorithms to be considered as alternatives for the top left optimiser in Fig 6 are PSO[12], ALPSO[33], COBYLA[34], Nelder-Mead[35], SLSQP[36] and CONMIN[37].

Table 1. Set of tools (models) available to every MDAO module.

Module	Tools available
Downstream wake effects	**1**. Jensen[18] | **2**. Larsen[19] | **3**. Ainslie 1D[20] | **4**. Ainslie 2D[21] | Each tool is furthermore instantiated with a range of number of wind sectors considered and number of bins into which the Weibull distributions are discretised.
Wake merging	**1**. Root sum square | **2**. maximum deficit | **3**. deficit product | **4**. deficit sum.
Wind turbine performance	**1**. Constant thrust coefficient and power | **2**. WindSim (simple BEM)[22] | **3**. WT_Perf (BEM with corrections)[23] | **4**. FAST (BEM with corrections, aero-hydro-servo-elastic simulation)[24].
Wake turbulence	**1** Constant turbulence | **2** Frandsen 2[25] | **3** Danish Recommendation[26] | **4** Frandsen 1[27] | **5** Larsen[28] | **6** Quarton[29].
Infield cable topology	**1** Constant cost | **2** Esau-Williams heuristic algorithm[30,31] | **3** Radial topology | **4** Random topology.
Support structure design	**1** Constant support structure cost | **2** TeamPlay[32].

IV.C. Phase 1: selection of analysis blocks

We start off by defining the criteria to evaluate the multi-dimensional performance of the MDAO analysis blocks. With optimisation at the heart of our use case, we desire fast analysis blocks that allow the exploration of many designs. It is also of interest that they have high accuracy to provide realistic optimal designs. In addition, we look at the concept of precision, to ensure our analysis block yields consistent results.

The metrics associated with these criteria are:

- Accuracy: We compare the absolute difference of the output of all alternatives against a referent. The levelised cost of energy of a real wind farm is a figure hard to come by for two reasons: the true costs of operation, maintenance and decommissioning are needed and these are only known at the end of the lifetime of the wind farm, and financial and production figures are seldom released to the public. These reasons lead us to benchmark the MDAO analysis blocks with respect to the particular alternative that is considered by the authors to be the "most sophisticated".

  One can never ascertain the absolute accuracy of a simulation tool, since its use usually lies in regions of the domain where the tool was not validated and calibrated. Instead, the user's confidence can only increase with the number of benchmark tests passed. Therefore, we propose to consider a number of different designs to analyse and then calculate the average of the difference with the referent:

  $$C_{acc}(W_j) = \frac{1}{n}\sum_{i=0}^{n} \left|(LCOE(W_j^i) - LCOE(W_r^i)\right| \tag{1}$$

  where W_j^i is the analysis block j used for simulating design i, n is the number of designs evaluated and C_{acc} is our metric for accuracy. The lower the value of the metric, the more accurate that analysis block will be.

- CPU time (C_{time}): Execution time can be measured in absolute terms and compared directly. The only consideration is to time each analysis more than once in order to reduce the impact from other processes running in the CPU simultaneously.

- Precision ($C_{precision}$): Due to the existing randomness in some tools, we can expect a spread in the LCOE estimated for a single design with the same analysis block. We thus aim to characterise the precision of the output. We use the standard deviation as a metric for this criterion. More precise analysis blocks score lower in $C_{precision}$.

In terms of the attributes any list of criteria should comply with, ours is deemed complete for our purpose of testing this guideline, as it captures the trade-off between time and accuracy and detects those alternatives that yield deterministic results. The list is also justified to be operative, as it picks up differences and metrics are measurable and meaningful, small and decomposable (these are taken from the criteria tree in Fig. 4). Lastly, non-redundancy is analysed with a plot matrix (Fig 7) made by randomly sampling the space of alternatives and measuring all three criteria per sample. We also provide pairwise Pearson correlation coefficients. The matrix plot shows that the metrics for these criteria are pairwise independent, a notion further strengthened by the correlation coefficients being close to zero. A correlation coefficient close to 1 or -1 would, on the contrary, suggests a linear dependence and thus further action would have to be taken. This might entail discarding one of the criteria or combining them into a single objective.

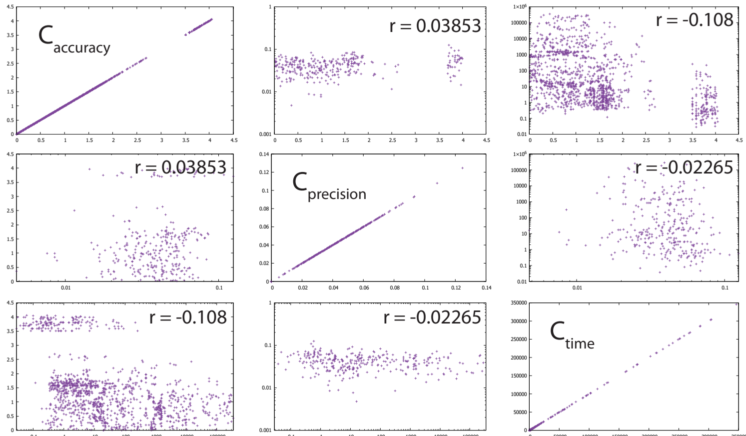

Figure 7. Plot matrix of the criteria scores by sampling random MDAO analysis blocks and the Pearson correlation coefficients.

Now that we have a list of criteria, we may proceed with the optimisation of the analysis block's tools by running MOPSOC. The formulation of this optimisation problem is:

$$\underset{W}{\text{minimize}} \quad C_{precision}(W),\ C_{time}(W),\ C_{accuracy}(W), \tag{2}$$

and we enforce constraints to the time and accuracy criteria *a posteriori*. However, these constraints could have been included in the formulation of the optimisation problem as well.

Figure 8 shows the approximated Pareto front found by MOPSOC, and the set of non-dominated MDAO analysis blocks is summarised in Fig. 9. Five alternatives, highlighted in red, are either too computationally expensive for an optimisation or have an unacceptably high error. Seven candidates are consequently selected for phase 2.

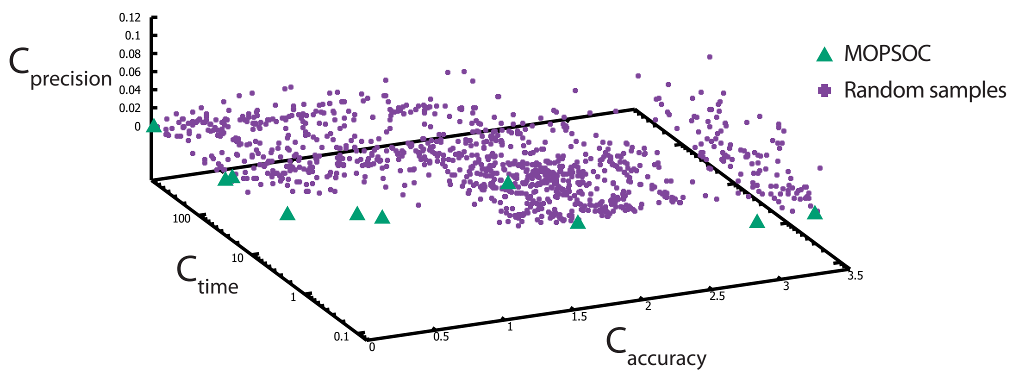

Figure 8. Plot of the scores of a sample of MDAO analysis blocks and the approximated Pareto front found by the Multi-Objective Particle Swarm Optimisation algorithm for Categorical variables (MOPSOC).

Every point on the approximated Pareto front corresponds to the scores of one analysis block.

Candidate analysis block	No. wind bins	No. wind sectors	Wake model				Wake merge				Turbine				Turbulence							Collector				Support		C_{acc}	C_{time}	C_{prec}
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	1	2	3	4	1	2			
1	7	36	●						●		●					●						●				●		3.362	0.06	0.000
2	12	36		●			●					●				●							●				●	0.042	29.70	0.003
3	3	36	●						●		●						●								●		●	1.299	0.34	0.052
4	4	72	●						●			●			●							●				●		0.648	1.67	0.000
5	7	72	●				●					●				●						●					●	0.008	32.17	0.000
6	8	72		●					●			●							●			●				●		0.104	3.85	0.000
7	4	72			●			●					●		●								●			●		0.003	773.54	0.000
8	8	36			●		●						●					●				●					●	0.006	726.87	0.000
9	12	36	●						●		●				●							●					●	1.785	0.30	0.000
11	6	36	●						●		●					●							●			●		2.877	0.09	0.000
10	4	36	●					●			●				●									●		●		3.294	0.09	0.000
12	6	72	●						●		●				●									●		●		0.521	2.34	0.000

Figure 9. Tools in the MDAO analysis blocks of the approximated Pareto front and their scores. In red we highlight the analysis blocks that will not be considered in phase 2. The tool number refers to Table 1.

IV.D. Phase 2: selection of driver algorithms

The seven analysis blocks selected in phase 1 are now coupled to the alternative wind farm layout optimisation algorithms.

The criteria and metrics with which we evaluate the performance of MDAO workflows are:

- Optimality: criterion that expresses the absolute performance of the best design found during the optimisation. Its metric ($C_{optimality}$) is the average of the objective function evaluation of the optimal solutions found by several identical runs.
- Feasibility: optimisation algorithms may yield unfeasible results, and thus those optimal designs may be accepted and slightly modified by the designer. We measure feasibility ($C_{feasibility}$) by finding the sum of the absolute distances between the boundaries of the farm and the wind turbines placed outside the confined area.
- Precision: this criterion measures the spread of the optimality. A wind plant designer has preference for algorithms that yield the same optimum consistently. A metric for this criterion ($C_{precision}$) is the standard deviation of the objective function evaluation of the optimal solution obtained by several identical optimisation runs.

We apply the ϵ-non-dominated sorting algorithm to all the scores of all alternatives to find the non-dominated set of available MDAO workflows.

The non-dominated alternatives are shown in Fig. 10 with their values normalised with the respect to the maximum and minimum across the Pareto front. Ranking alternatives based on their distance to the utopia point is only meaningful with normalised criteria, to avoid the scales of the metrics from inducing any bias.

V. Validation of this guideline

The top ranking MDAO workflows result from coupling the Nelder-Mead and PSO algorithms to the 5^{th} analysis block in Fig. 9. The most useful analysis is composed of the Jensen wake model, root sum square wake merging, simple BEM, Frandsen 2 turbulence model, a constant cost for the electrical collection system and the TeamPlay module for support structure design. The combination of these tools resulted in an accurate, though slower analysis block than the rest of the alternatives in the approximated Pareto front. The only surprise was the inclusion of the constant electrical collection cost, though it can be explained by the fact that the output was calibrated using an equally spaced design, and the difference in cable lengths between optimal layouts and the baseline had less impact on LCOE than did water depth and power production. This is especially true in a layout with only a few wind turbines. It is also worth discussing the fact that the accuracy of *higher fidelity* tools such as Ainslie and FAST do not justify their use in an early stage layout optimisation, due to their low speed.

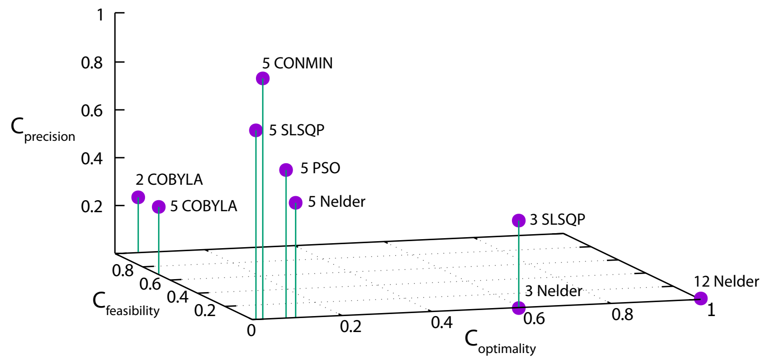

Figure 10. Pareto front of MDAO workflows with their scores mapped to a range from 0 to 1. Analysis block number 5 coupled to the Nelder-Mead and PSO optimisation algorithms are the closest to the utopia point.

The Nelder-Mead method scores high in feasibility and precision, meaning it consistently reaches the same optimal values without violating constraints. We acknowledge that the optimality metric in this work does not represent the true performance of the optimisers, as it is a value closely tied to the accuracy of the analysis block. Instead, a good alternative for this criterion would be the absolute improvement of the objective function between the first and last iterations.

Concerning the criteria for phase 1, we saw the need to include a criterion for the sensitivity of tools with respect to the input parameters. The accuracy criterion picked up the deficient performance of the constant power and constant turbulence tools, yet an additional criterion for sensitivity would have helped discard the constant support structure cost. In this case study, it is rather obvious that those constant output tools should never be used in a layout optimisation where we aim to capture the effect of wakes, water depth and distance between turbines. However, in other cases where tools may be black boxes, there might not be enough information regarding which input variables affect the outputs.

An improvement to consider next is to evaluate the sensitivity of the rankings. Differences may arise due to differences in the definition of the metrics for the criteria, the spread of the values of the metrics, the optimisation formulations for the MOPSOC algorithm, and the completeness of the list of criteria.

VI. Conclusion

The motivation for this research is based on our observation that researchers increasingly apply MDAO in the field of wind energy disregarding what model fidelity and driver algorithm should be coupled in the workflow.

We present a guideline for evaluating, comparing, ranking and selecting the best performing MDAO workflows for a predefined use case. This guideline consists of two phases. First, we treat the selection of the most useful MDAO analysis blocks as a multi-objective optimisation problem. We provide guidelines for selecting the criteria that describe the overall performance of the analysis blocks.

The second phase involves coupling the best candidates found in phase 1 to a set of driver algorithms. The best performing MDAO workflows are found by scoring all combinations against multiple criteria and finding the Pareto front using the ϵ-non-dominated sorting algorithm.

We provide an example of the application of this guideline to a common problem in the field of wind energy, the multidisciplinary optimisation of the layout of an offshore wind plant. By following the guideline, the need to solve the trade-offs typically found when dealing with the choice of model fidelity are further supported by the results: the approximated Pareto front for the analysis blocks (Fig. 8) covers a wide range in the accuracy and time criteria. This shows that there is not an obvious boundary between the best and worst alternatives. In particular we see that the Jensen wake model and a simple blade element momentum model perform better than more sophisticated tools in an early stage of the design process.

We conclude, additionally, that qualitative reasoning is not always enough to guarantee a useful criteria tree. This is evidenced by the fact that the variability in the precision criterion is less than that of time and accuracy, and provides thus less discriminating power. Instead, the criteria tree building process can be informed by a qualitative pre-assessment of criteria, e.g. testing the range of scores for a sample of alternatives.

Our guideline provides a set of better performing MDAO workflows to be considered by the user. While a weighted aggregation approach yields a single best performing alternative, ours allows the user to make an informed decision with the information it supplies. The user may now choose an alternative that performs better for a given criterion with limited loss of performance with respect of the other criteria.

Extending the case study to include higher fidelity tools and other key criteria is envisioned in future work.

Acknowledgements

This research is funded by the Mexican National Council for Science and Technology (CONACYT).

References

[1]Sanchez Perez-Moreno, S., Zaaijer, M., Bottasso, C., Dykes, K., Merz, K., Réthoré, P.-E., and Zahle, F., "Roadmap to the multidisciplinary design analysis and optimisation of wind energy systems," *Journal of Physics: Conference Series*, Vol. 753, IOP Publishing, 2016, p. 062011.

[2]Dykes, K., Rethore, P., Zahle, F., and Merz, K., "IEA Wind Task 37 Final Proposal. Wind Energy Systems Engineering: Integrated RD&D." Tech. rep., International Energy Agency, 2015.

[3]Belton, V. and Stewart, T., *Multiple Criteria Decision Analysis: An Integrated Approach*, Springer US, 2012.

[4]Gross, D., "Report from the Fidelity Definition and Metrics Implementation Study Group (FDM-ISG). Orlando, Florida: Simulation Interoperability Standards Organization (SISO); 1999 March. Report No," Tech. rep., 99S-SIW-167, 1999.

[5]Harmon, S. Y. and Youngblood, S., "Leveraging fidelity to achieve substantive interoperability," *Proceedings of the 2001 Spring Simulation Conference*, 2001, pp. 25–30.

[6]Roza, Z. C., *Simulation fidelity theory and practice*, Ph.D. thesis, Technische Universiteit Delft, 2005.

[7]Pace, D. K., "Fidelity, Resolution, Accuracy, and Uncertainty," *Modeling and Simulation in the Systems Engineering Life Cycle*, Springer, 2015, pp. 29–37.

[8]Keeney, R. L. and Raiffa, H., *Decisions with multiple objectives: preferences and value trade-offs*, Cambridge university press, 1993.

[9]Keeney, R. L. and Raiffa, H., "Decision with multiple objectives," 1976.

[10]Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," *International Conference on Parallel Problem Solving From Nature*, Springer, 2000, pp. 849–858.

[11]Sanchez Perez-Moreno, S. and Zaaijer, M. B., "Multiobjective Particle Swarm Optimisation for Categorical Variables," (to be submitted).

[12]Eberhart, R. C. and Kennedy, J., "A new optimizer using particle swarm theory," *Proceedings of the sixth international symposium on micro machine and human science*, Vol. 1, New York, NY, 1995, pp. 39–43.

[13]Hassan, R., Cohanim, B., De Weck, O., and Venter, G., "A comparison of particle swarm optimization and the genetic algorithm," *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2005.

[14]Das, I., "A preference ordering among various Pareto optimal alternatives," *Structural and Multidisciplinary Optimization*, Vol. 18, No. 1, 1999, pp. 30–35.

[15]Laumanns, M., Thiele, L., Deb, K., and Zitzler, E., "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary computation*, Vol. 10, No. 3, 2002, pp. 263–282.

[16]Jonkman, J., Butterfield, S., Musial, W., and Scott, G., "Definition of a 5-MW reference wind turbine for offshore system development," Tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO., 2009.

[17]Lambe, A. B. and Martins, J. R., "Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 273–284.

[18]Katic, I., Højstrup, J., and Jensen, N., "A simple model for cluster efficiency," *European Wind Energy Association Conference and Exhibition*, 1986, pp. 407–410.

[19]Larsen, G. C., "A simple wake calculation procedure," Tech. rep., Riso National Laboratory, 1988.

[20]Anderson, M., "Simplified Solution to the Eddy-Viscosity Wake Model," *Document prepared by Renewable Energy Systems Ltd (RES)*, 2009.

[21]Ainslie, J. F., "Calculating the flowfield in the wake of wind turbines," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 27, No. 1, 1988, pp. 213–224.

[22]van Dommelen, D., *Pythons wind turbine design package manual*, Delft University of Technology, September 2013.

[23]Buhl, M., "NWTC Design Codes (WT-Perf)," http://wind.nrel.gov/designcodes/simulators/wtperf, 2008, accessed July-2008.

[24]Jonkman, J., "The new modularization framework for the FAST wind turbine CAE tool," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 202.

[25]Frandsen, S. T., "Generated structural loading in wind turbine clusters," 2007.

[26]"Recommendation for the fulfillment of the requirements found in the technical criteria." Tech. rep., Danish Energy Agency, 1992.

[27]Frandsen, S. T., Chacón, L., Crespo, A., Enevoldsen, P., Gómez-Elvira, R., Hernández, J., Højstrup, J., Manuel, F., and Thomsen, K., "Measurements on and modelling of offshore wind farms," Tech. rep., 1996.

[28]Larsen, G. C., Højstrup, J., and Madsen, H. A., "Wind fields in wakes," 1996.

[29]Quarton, D. and Ainslie, J., "Turbulence in wind turbine wakes," *Wind Engineering*, 1990, pp. 15–23.

[30]Esau, L. R. and Williams, K. C., "On teleprocessing system design, Part II: A method for approximating the optimal network," *IBM Systems Journal*, Vol. 5, No. 3, 1966, pp. 142–147.

[31]Maselis, A., *Layout Optimization of Offshore Wind Farms affected by Wake effects, Cable topology and Support Structure variation*, Master's thesis, Delft University of Technology, 2016.

[32]Zaaijer, M. B., *Great expectations for offshore wind turbines: Emulation of wind farm design to anticipate their value for customers*, Ph.D. thesis, Delft University of Technology, 2013.

[33]Sedlaczek, K. and Eberhard, P., "Using augmented Lagrangian particle swarm optimization for constrained problems in engineering"¿ Using augmented Lagrangian particle swarm optimization for constrained problems in engineering," *Structural and Multidisciplinary Optimization*, Vol. 32, No. 4, 2006, pp. 277–286.

[34]Powell, M. J., "A direct search optimization method that models the objective and constraint functions by linear interpolation," *Advances in optimization and numerical analysis*, Springer, 1994, pp. 51–67.

[35]Nelder, J. A. and Mead, R., "A simplex method for function minimization," *The computer journal*, Vol. 7, No. 4, 1965, pp. 308–313.

[36]Kraft, D., "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[37]Vanderplaats, G. N., "CONMIN: a Fortran program for constrained function minimization: user's manual," 1973.