Optimisation and Multidisciplinary Design Analysis\&Optimisation
From Aerospace to Current and Potential Activities in the Field of Hemodynamics

Dr Edmondo Minisci
Department of Mechanical and Aerospace Engineering
University of Strathclyde, Glasgow, Scotland, UK
Outline

- Introduction
- Optimization
  - Introduction & OR
  - Single-Objective Problems
  - Computational Intelligence
  - Multi-Objective Problems
  - Final Remarks
- Multidisciplinary Design Analysis and Optimization
  - Architectures
  - Re-Entry Test Case
  - MDAO of Medical Devices
  - Hemodynamics at Strathclyde
Optimal Control

The design of complex engineering systems requires the efficient integration of multiple disciplines.

The design should be optimal with respect of a number of criteria and fulfil a set of constraints.

The design should be robust:
- Product development within expected margins
- Reliable product behaviour during life-time

Shape and Structure
Aerodynamics and Thermal

FESTIP - Concept FSS-5
M = 5 alpha=7.5 delta=20

Mach number
- 3.7793
- 5.4259
- 5.1971
- 4.8384
- 4.6146
- 4.3904
- 4.1621
- 3.7358
- 3.4137
- 3.1780
- 2.8473
- 2.5067
- 2.2386
- 2.9211
- 1.7383
- 1.4465
- 1.1542
- 0.8611
- 0.5774
- 0.2897

Introduction
Optimization
Intro & OR
Comp. Intell.
MOP
Final Remarks
MDAO
Architectures
Re-Entry Case
Med. Devices
Strath. Hemo
Integrated vs. Non-integrated Approach

System Design

Non-Integrated Approach

Monte Carlo Analysis

Trajectory

FESTIP - Concept FSS-5
M = 5 alpha=7.5 delta=20
Integrated vs. Non-integrated Approach

Integrated Approach

System Level
Multicriteria Optimisation

Introduction
Optimization
Intro & OR
Comp. Intell.
MOP
Final Remarks
MDAO
Architectures
Re-Entry Case
Med. Devices
Strath. Hemo
Integrated vs. Non-integrated Approach

- An integrated approach **saves** development time
- An integrated approach produces an **optimal result** at system and subsystem level with respect to multiple criteria
- An integrated approach provides a **reliable** solution with a prediction of the expected system margins
- Ultimately **reduces** production **cost** and improves performance
Multidisciplinary Design Optimisation

- **Environment**
  - HPC facilities
  - Efficient models
  - Meta-modelling

- **MDO approaches**
  - Decomposition/Aggregation
  - Collaboration
  - Evolution-control
  - Concurrency

- **Optimisation**
  - Deterministic/Stochastic
  - Multi-objective
  - Constraints

- **Uncertainty**
  - Structure representation
  - Quantification

- **Multi-fidelity**
OPTIMIZATION
Introduction

Optimization

• Optimization is derived from the Latin word “optimus”, the best.

• Optimization characterizes the activities involved to find “the best”.

• People have been “optimizing” forever, but the roots for modern day (engineering) optimization can be traced to the Second World War.
Operational Research

• Operational Research originated from the activities performed by multidisciplinary teams formed in the British armed forces involved in solving complex strategic and tactical problems in World War II.

• Waddington describes the main objectives of the Operational Research Section in the British armed forces as

“The prediction of the effects of new weapons and tactics.”

Motivation for Operational Research

- Many problems associated with the Allied military effort were simply too complicated to expect adequate solutions from a single individual, or even a single discipline.

- All persons selected were talented (wo)men + wartime pressure + synergism generated from the interactions of different disciplines

- Due to their success, other allied nations adopted the same approach.
Because the work assigned to these groups were in the nature of military operations their work became known as *operational research* in the United Kingdom and as *operations research* in the United States.

The abbreviation OR is commonly used for both operational research and operations research.

Wartime examples: radar deployment, anti-aircraft fire control, fleet convoy sizing, submarine detection.
Operational Research was defined by the Operational Research Society of Great Britain as follows:

"Operational research is the application of the methods of science to complex problems arising in the direction and management of large systems of men, machines, materials and money in industry, business, government, and defense. The distinctive approach is to develop a scientific model of the system, incorporating measurements of factors such as chance and risk, with which to predict and compare the outcomes of alternative decisions, strategies or controls.

The purpose is to help management determine its policy and actions scientifically."
• “Operational Research ("OR"), also known as Operations Research or Management Science ("OR/MS") looks at an organisation's operations and uses mathematical or computer models, or other analytical approaches, to find better ways of doing them.”

• See www.orsoc.org.uk
After the War...

- Many of the scientists in the OR groups turned their activities to applying their approach to civilian problems.

- Some returned to universities to develop a sound foundation for the hastily developed techniques, others concentrated on developing new techniques.

- First civilian organizations interested were large profit making corporations. For example, petroleum companies were the first to use linear programming on a large scale for production planning. First only big business could afford it.

- Applications in the service industries did not start until the mid 1960s.
It is generally accepted that without computers, OR and optimization would not be what they are today.

- Earlier mathematical models (such as calculus, Lagrange multipliers) relied on sophistication of technique to solve the problem classes for which they were suited.

- Methods of mathematical optimization (e.g., Linear Programming) rely far less on mathematical sophistication than they do on an unusual “adaptibility to the mode of solution inherent in the modern digital computer”.

- Particularly striking is the simplicity of these methods of mathematics coupled with their iterative processes (i.e., the repeated performance of a relatively simple set of operations)
Consider the case of Linear Programming (more later):

- The first large scale computer became a practical reality in 1946 at the University of Pennsylvania. This was just one year before the development of simplex.

- The simplex method for linear programming consists only of a few steps and these steps require only the most basic mathematical operations which a computer is well suited to handle.

- However, these steps must be repeated over and over before one finally obtains an answer.

- The first successful computer solution of a LP problem was in January 1952 on the National Bureau of Standards SEAC computer.
You will often hear the phrase “**programming**” as in:

- mathematical programming,
- linear programming,
- nonlinear programming,
- mixed integer programming, etc.
By the way ... (cont.)

This has (in principle) **nothing** to do with modern day computer programming.

In the early days, a **set of values** which represented a solution to a problem was referred to as a “**program**”.

Nowadays you program (software) to find a program!
Type of Optimisation Problems

- Single- / Multi-Objective
- Continuous / Discrete / Mixed
- Deterministic / Stochastic
- Constrained / Unconstrained
- Linear / Non-Linear
SYSTEM

\[ f(\mathbf{x}) \]

\[ g(\mathbf{x}) \]

Response of the system only function of design parameters \( \mathbf{x} \)
Single-Objective Optimisation

A single-objective optimization problem has the form

$$\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq b_i, \quad i = 1, 2, \ldots, m
\end{align*}$$

or, in other form

$$\begin{align*}
\text{subject to} & \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, m
\end{align*}$$
A single-objective optimization problem has the form

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq b_i, \quad i = 1, 2, \ldots, m
\end{align*}
\]

or, in other form

\[
\begin{align*}
\text{subject to} & \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, m
\end{align*}
\]

where the vector \( x = [x_1, x_2, \ldots, x_d]^T \) is the optimization variable vector of the problem, the function \( f_0 : \mathcal{R}^d \to \mathcal{R} \) is the objective function (or cost function), the functions \( f_i : \mathcal{R}^d \to \mathcal{R}, \quad i = 1, 2, \ldots, m, \) are the (inequality) constraint functions, and the constants \( b_1, b_2, \ldots, b_m \) are the limits, or bounds, for the constraints. A vector \( x^* \) is called optimal, or a solution of the problem 2.1, if it has the smallest objective value among all vectors that satisfy the constraints: for any \( x \) with \( f_1(x) \leq b_1, f_2(x) \leq b_2, \ldots, f_m(x) \leq b_m \), we have \( f_0(x) \geq f_0(x^*) \) (Constraints define the sub-set \( S \) and \( x^* \in S \subseteq \mathcal{R}^d \)).
Linear Programming

objective and all constraint functions are linear:

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i, \quad i = 1, 2, \ldots, m
\end{align*}
\]

where the vectors \( c, a_1, \ldots, a_m \in \mathcal{R}^d \) and scalars \( b_1, b_2, \ldots, b_m \in \mathcal{R} \) are problem parameters that specify the objective and constraint functions.
Solving the LP problems: the Simplex

- A large variety of Simplex-based algorithms exist to solve LP problems.

- Other (polynomial time) algorithms have been developed for solving LP problems:
  - Khachian algorithm (1979)
  - Kamarkar algorithm (AT&T Bell Labs, mid 80s)
  - See Section 4.10

BUT,

none of these algorithms have been able to beat Simplex in actual practical applications.

HENCE,

Simplex (in its various forms) is and will most likely remain the most dominant LP algorithm for at least the near future
Solving the LP problems: Fundamental Theorem

Extreme point (or Simplex filter) theorem:

If the maximum or minimum value of a linear function defined over a polygonal convex region exists, then it is to be found at the boundary of the region.

Convex set:

A set (or region) is convex if, for any two points (say, \(x_1\) and \(x_2\)) in that set, the line segment joining these points lies entirely within the set.

A point is by definition convex.
What does the extreme point theorem imply?

• A finite number of extreme points implies a finite number of solutions!

• Hence, search is reduced to a finite set of points

• However, a finite set can still be too large for practical purposes

• Simplex method provides an efficient systematic search guaranteed to converge in a finite number of steps.
Basic Steps of Simplex

1. Begin the search at an extreme point (i.e., a basic feasible solution).

2. Determine if the movement to an adjacent extreme can improve on the optimization of the objective function. If not, the current solution is optimal. If, however, improvement is possible, then proceed to the next step.

3. Move to the adjacent extreme point which offers (or, perhaps, appears to offer) the most improvement in the objective function.

4. Continue steps 2 and 3 until the optimal solution is found or it can be shown that the problem is either unbounded or infeasible.
Convex Optimization

A convex optimization problem is one of the form

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq b_i, \quad i = 1, 2, \ldots, m
\end{align*}
\]  

(2.5)

where the functions \( f_0, f_1, \ldots, f_m : \mathcal{R}^d \rightarrow \mathcal{R} \) are convex, i.e., satisfy

\[
f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)
\]

(2.6)
A convex optimization problem is one of the form

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq b_i, \quad i = 1, 2, \ldots, m
\end{align*}
\] (2.5)

where the functions \( f_0, f_1, \ldots, f_m : \mathcal{R}^d \to \mathcal{R} \) are convex, i.e., satisfy

\[
f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)
\] (2.6)

for all \( x, y \in \mathcal{R}^d \) and all \( \alpha, \beta \in \mathcal{R} \) with \( \alpha + \beta = 1, \alpha, \beta \geq 0 \). Comparing 2.4 and 2.5, we see that convexity is more general than linearity: inequality replaces the more restrictive equality. Since any linear program is therefore a convex optimization problem, we can consider convex optimization to be a generalization of linear programming.
When, for an optimization problem, the objective and/or constraint functions are not linear, but not known to be convex, it is usually pointed out as *nonlinear optimization* (or *nonlinear programming*). There are no effective methods for solving general nonlinear programming problems. Even simple looking problems with few variables can be extremely challenging, while problems with a few hundreds of variables can be intractable.

Generally speaking, there are two philosophies to handle non linear problems:

a) *local search* (or *local optimization*), and

b) *global search* (or *global optimization*).
Nonlinear Optimization

When, for an optimization problem, the objective and/or constraint functions are not linear, but not known to be convex, it is usually pointed out as nonlinear optimization (or nonlinear programming).

There are no effective methods for solving general nonlinear programming problems. Even simple looking problems with few variables can be extremely challenging, while problems with a few hundreds can be intractable.

Generally speaking, there are two philosophies to handle nonlinear problems:

a) local search (or local optimization), and
b) global search (or global optimization).

Constraint actually (meaning).

A few in a few a linear
Nonlinear Optimization

Local Optimization

- In local optimization, the compromise is to give up seeking the optimal $x$, which minimizes the objective over all feasible points. Instead we seek a point that is only locally optimal, which means that it minimizes the objective function among feasible points that are near it, but is not guaranteed to have a lower objective value than all other feasible points.

- A large fraction of the research on general nonlinear programming has focused on methods for local optimization, which as a consequence are well developed.
Nonlinear Optimization

Local Optimization

• In local optimization, the compromise is to give up seeking the optimal $x$, which minimizes the objective over all feasible points. Instead we seek a point that is only locally optimal, which means that it minimizes the objective function among feasible points that are near it, but is not guaranteed to have a lower objective value than all other feasible points.

• A large fraction of the research on general nonlinear programming has focused on methods for local optimization, which as a consequence are well developed.
Nonlinear Optimization

Local Optimization

• There are some disadvantages of local optimization methods, beyond (possibly) not finding the true, globally optimal solution.
  
  – The methods require an initial guess for the optimization variable. This initial guess or starting point is critical, and can greatly affect the objective value of the local solution obtained.
  
  – Moreover, little information is provided about how far from (globally) optimal the local solution is.
Nonlinear Optimization

Global Optimization

- In global optimization, the true global solution of the optimization problem is found;
- The compromise is on the efficiency. (In the worst-case, complexity of global optimization methods grows exponentially with the problem sizes $d$ and $m$)
- Traditional techniques of global optimization, such as a) branch and bound, b) multi start algorithms, and c) clustering methods, are usually used for problems with a small number of variables, where computing time is not critical, and the value of finding the true global solution is very high.
Zero and first order algorithms

- You often must choose between algorithms which need only evaluations of the objective function or methods that also require the derivatives of that function.

- Algorithms using derivatives are generally more powerful, but do not always compensate for the additional calculations of derivatives.

- Note that you may not be able to compute the derivatives.
Basic Descent Methods

- Basic descent methods are the basic techniques for iteratively solving unconstrained minimization problems.

- Important for practical situations because they offer the simplest and most direct alternatives for obtaining solutions.

- Also good as a benchmark.
General Basic Descent Method Algorithm

Basic steps:
1. start at an initial point;
2. determine according to a fixed rule a direction of movement; and
3. move in that direction to a (relative) minimum of the objective function on that line.
4. At the new point, a new direction is determined and the same process is repeated.

- The primary difference between algorithms (steepest descent, Newton's method, etc) is the rule by which successive directions of movement are selected.
Computational Intelligence

• Computational Intelligence is NOT Artificial Intelligence

• In general, typical Artificial Intelligence techniques are strongly oriented to symbolic representations and manipulations (reasoning) in a *top-down* manner.
  – the structure of a given problem (environment, domain context) is analyzed beforehand and the construction of an intelligent system is based upon this structure.

• Computational intelligence techniques are generally *bottom-up*, where order and structure emerge from an unstructured beginning, where the only knowledge is “some sort” of numerical data-base.
Computational Intelligence

• Computational Intelligence is “low-level computation in the style of the mind”, whereas Artificial Intelligence is “mid-level computation in the style of the mind”.
  – The difference is that mid-level systems include knowledge, while low-level systems do not.

• The areas covered by the term computational intelligence are also known under the name soft computing.

• This name was chosen to indicate the difference between soft computing and operations research, also known as hard computing.
Computational Intelligence

- Soft and Hard computing are connected by the problem domains they are applied in, but
- while operations research algorithms usually come with crisp (and strict) conditions on the scope of applicability and proven guarantees for a solution (or even an optimal solution),
- soft computing puts no conditions on the problem but also provides *no guarantees for success*, a deficiency which is compensated by the robustness of the methods.
Computational Intelligence

- Requirement for little, if any, *a priori* knowledge relating to the problem.
- Excellent exploratory capabilities.
- Ability to avoid local optima.
- Ability to handle high dimensionality.
- Robustness across a wide range of problem classes.
Simulated Annealing

- In 1983, Kirkpatrick and co-workers proposed a method of using a Metropolis Monte Carlo simulation to find the lowest energy (most stable) orientation of a system.
- They called the method Simulated Annealing because it is inspired by the annealing process of metals during cooling.
- In short the annealing can be described as follows: at high temperatures the molecules in a metal move freely but as the metal is cooled this movement is gradually reduced and atoms align to form crystals. The crystalline form constitutes a state of minimum energy.
- Metals that are cooled gradually reach a state of minimum energy naturally, while if they are forcibly cooled they reach a polycrystalline or amorphous state of which energy level is much higher.
- Annealed metals have better mechanical characteristics.
Computational Intelligence

- Evolutionary Algorithms (EAs) are stochastic search methods that take their inspiration from natural selection and survival of the fittest in the biological world.
- By analogy to natural evolution, the solution candidates are called *individuals* and the set of solution candidates is called the *population*.
- Each individual represents a possible solution, i.e., a decision vector, to the problem at hand, or encodes it based on an appropriate representation.
Computational Intelligence

• EA is a stochastic approach which:
  • a) principally *memorizes* a population of solutions;
  • b) has some kind of *mating selection*;
  • c) has some kind of recombination and mutation as *variation operators*; and
  • d) has some kind of *environment selection*.
Computational Intelligence

- EA is a stochastic approach which:
- a) principally *memorizes* a population of solutions;
- b) has some kind of *mating selection*;

\[
\begin{array}{c}
\text{Information: Population} \\
\text{Selection: from Environment} \\
\text{Variation: Crossover & Mutation} \\
\text{Selection: for Mating}
\end{array}
\]
Computational Intelligence

Estimation of Distribution Algorithms (EDAs)

- If the mimicking of natural evolution in living species has been a source of inspiration of new strategies, the attempt to copy natural techniques as they are sometimes introduces a great complexity without a corresponding improvement of algorithms performance.

- Moreover standard evolutionary algorithms can be ineffective when problems exhibit a high level of interaction among variables. This is mainly due to the fact that recombination operators are likely to disrupt promising sub-structures of optimal solutions.
Computational Intelligence

Estimation of Distribution Algorithms (EDAs)

- Some algorithms have been proposed that automatically learn the structure of the search space.
- Instead of implicit reproduction of important building blocks and their mixing by selection and two-parent recombination operators, new solutions are generated by using the information extracted from the entire set of promising solutions.
- Starting from results of current populations, these methods try to identify a probabilistic model of the search space, and crossover and mutation operators are replaced with sampling.
Estimation of Distribution Algorithms (EDAs)

- Some algorithms have been proposed that automatically learn the structure of the search space.

Information: Population

Selection: from Environment

Selection: for Learning

Variation: Probabilistic Modelling and Sampling
MULTI-OBJECTIVE OPTIMIZATION
Multi-Objective Optimisation

find the vector \( \mathbf{x}^* = [x_1^*, x_2^*, \ldots, x_d^*]^T \in S \subseteq \mathbb{R}^d \) that will satisfy the \( m \) constraints:

\[
g_i (\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, m
\]  

(2.7)

and optimize the vector function

\[
\mathbf{f} (\mathbf{x}) = [f_1 (\mathbf{x}), f_2 (\mathbf{x}), \ldots, f_k (\mathbf{x})]^T \in \mathcal{F} \subseteq \mathbb{R}^k
\]  

(2.8)

we wish to determine from among the set \( S \) of all numbers that satisfy (2.7) the particular solution \( \mathbf{x}^* \) that yields the optimum values of all the objective functions.

The vector function \( \mathbf{f}(\mathbf{x}) \) is a function that maps the set \( S \) in the set \( F \) which represents all possible values of the objective functions. The \( k \) components of the vector \( \mathbf{f}(\mathbf{x}) \) represent the non-commensurable criteria that must be considered.
MOO: Ideal Point

The problem is that the meaning of optimum is not well defined in this context, since we rarely have an $x^*$ such that for all $i = 1, 2, \ldots, k$

$$\forall x \in S \ [f_i(x^*) \leq f_i(x)]$$

(2.9)

(a) Particular case of a One Point solution  
(b) Ideal point
Multi-Objective Optimisation

The concept of a Pareto optimum was formulated by Pareto in the nineteenth century and by itself constitutes the origin of research in multi-objective optimization.

We say that a point \( \mathbf{x}^* \in S \) is Pareto optimal for a minimization problem if

\[
\text{there is no } \mathbf{x} \in S \text{ such that } f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*) \quad i = 1, \ldots, k
\]

\[
 f_i(\mathbf{x}) < f_i(\mathbf{x}^*) \quad \text{at least one } i
\]
Strictly tied to the concept of Pareto optimality is the concept of dominance: if $x^1, x^2 \in S$ and
\[
    f_i(x^1) \leq f_i(x^2) \quad i = 1, \ldots, k \\
    f_i(x^1) < f_i(x^2) \text{ at least one } i
\]
then we say that $x^1$ dominates $x^2$ and $f(x^1)$ dominates $f(x^2)$. 

![Graph showing the Pareto front with a shaded area and labeled axes](image-url)
MOO: Formulating the Objective

“How can we handle such a kind of problems?“
When the objectives are more than one it is required to chose an approach to discriminate different solutions.

Information:

Selection: for “surviving”

Variation:

Selection: for Variations
MOO: Formulating the Objective

- The multi-objective optimization problem (MOOP) can be handled in four different ways depending on when the user (or decision-maker) articulates the preferences concerning the different objectives:
  - never,
  - before,
  - during, or
  - after
- the actual optimization procedure.
MOO: Formulating the Objective

• In the first two approaches, the different objectives are aggregated to one overall objective function and the optimization is then conducted with one optimal design as the result. The optimization results are then strongly dependent on how the objectives were aggregated.

• The third approach is an iterative process where the decision-maker progressively articulates his preferences on the different objectives. The underlying assumption is that once the search for an optimal solution has started and the decision-maker has been presented with some alternatives, he will be better equipped to value the objectives.

• The most appealing is the fourth approach, where the optimization is conducted without articulating any preferences among the objectives. The outcome of this optimization is a set of Pareto optimal solutions which shows the trade-offs between the objectives. The user then has to trade the objectives against each other in order to select the final design. Thus, optimization is conducted before the decision-maker articulates his preferences.
MOO: Formulating the Objective

• In the first two approaches, the different objectives are aggregated to one overall objective function and the optimization is then conducted with one optimal design as the result. The optimization results are then strongly dependent on how the objectives were aggregated.

• The third approach is an iterative process where the decision-maker progressively articulates his preferences on the different objectives. The underlying assumption is that once the search for an optimal solution has started and the decision-maker has been presented with some alternatives, he will be better equipped to value the objectives.

• The most appealing is the fourth approach, where the optimization is conducted without articulating any preferences among the objectives. The outcome of this optimization is a set of Pareto optimal solutions which shows the trade-offs between the objectives. The user then has to trade the objectives against each other in order to select the final design. Thus, optimization is conducted before the decision-maker articulates his preferences.
MOO: Formulating the Objective

• In the first two approaches, the different objectives are aggregated to one overall objective function and the optimization is then conducted with one optimal design as the result. The optimization results are then strongly dependent on how the objectives were aggregated.

• The third approach is an iterative process where the decision-maker progressively articulates his preferences on the different objectives. The underlying assumption is that once the search for an optimal solution has started and the decision-maker has been presented with some alternatives, he will be better equipped to value the objectives.

• The most appealing is the fourth approach, where the optimization is conducted without articulating any preferences among the objectives. The outcome of this optimization is a set of Pareto optimal solutions which shows the trade-offs between the objectives. The user then has to trade the objectives against each other in order to select the final design. Thus, optimization is conducted before the decision-maker articulates his preferences.
MOO: Formulating the Objective

No Preference Articulation

- These are methods that do not use any preference information, and are known as compromise programming or goal programming (The term "goal programming" is used by its developers to indicate the search for an "optimal" program - i.e., a set of policies to be implemented - for a mathematical model that is composed solely of goals)

- "the ideal point can be used as reference point"

In general, given a distance measure $d_s : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{R}_+$, the compromise programming problem is

$$\text{minimize} \quad d_s (f(x) - f^0) \quad (2.15)$$

Particularly, when a $L_\infty$-metric is used we have the so called the Min – Max formulation, formalized as follows:

$$\text{minimize} \quad \|f(x) - f^0\|_\infty = \max_{i=1}^k |f_i(x) - f_i^0| \quad (2.16)$$
MOO: Formulating the Objective

A priori Articulation of Preference Information

• Weighted-sum Approaches

\[
\min \sum_{i=1}^{k} w_i f_i(x)
\]

where \( w_i \geq 0 \) are the weighting coefficients representing the relative importance of the objectives and it is usually assumed that

\[
\sum_{i=1}^{k} w_i = 1
\]
MOO: Formulating the Objective

**A priori Articulation of Preference Information**

- \( \varepsilon \)-constrained method

\[
\begin{align*}
\text{minimize} & \quad f_q(x) \\
\text{subject to} & \quad f_i(x) \leq \varepsilon_i, \quad i = 1, 2, \ldots, k, \quad i \neq q \\
& \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, m
\end{align*}
\]

where \( \varepsilon \in \mathbb{R}^{k-1} \).
MOO: Formulating the Objective

**A priori Articulation of Preference Information**

- **Lexicographic Approach**

In lexicographic optimization the lexicographic order when comparing objective vectors in criterion space is adopted. In this sense, for a minimization problem, a solution $\mathbf{f}^1$ is better than a solution $\mathbf{f}^2$, $\mathbf{f}^1 < \mathbf{f}^2$, if $f^1_q < f^2_q$ where $q = \min\{i : \mathbf{f}^1 \neq \mathbf{f}^2\}$.

- **Lexicographic optimality**, which guarantees Pareto optimality, implies a ranking of the objectives, for which the optimization of $f_1$ has the priority and the others objectives are optimized only in the case of multiple optimal solutions found for $f_1$. 
Progressive Articulation of Preference Information
• Generally referred to as interactive methods.
• These methods work according to the hypothesis that the user is unable to indicate preferences information \textit{a priori} due to the complexity of the problem. However, as the search moves on and the user learns more about the problem, he is capable of giving directions in which to look for improvements.

Advantages of these types of methods are:
\begin{enumerate}
\item there is no need for \textit{a priori} preference information;
\item only local preference information is needed;
\item it is a learning process where the user gets a better understanding of the problem;
\item as the user takes an active part in the search it is more likely that he accepts the final solution.
\end{enumerate}
MOO: Formulating the Objective

Progressive Articulation of Preference Information
• Generally referred to as interactive methods.
• These methods work according to the hypothesis that the user is unable to indicate preferences information \textit{a priori} due to the complexity of the problem. However, as the search moves on and the user learns more about the problem, he is capable of giving directions in which to look for improvements.

Advantages of these types of methods are:
   a) there is no need for \textit{a priori} preference information;
   b) only local preference information is needed;
   c) it is a learning process where the user gets a better understanding of the problem;
   d) as the user takes an active part in the search it is more likely that he accepts the final solution.
MOO: Formulating the Objective

Progressive Articulation of Preference Information

Disadvantages:

a) human effort is required during the whole search process;

These methods usually progress by either changing the weights in a weighted-sum approach, or by progressively reducing the search space. The decision-maker then has to determine a relaxation to some objectives in order to achieve improvements in others. The relaxed objectives are moved from the objective function and added as constraints to limit the solutions space.
MOO: Formulating the Objective

Progressive Articulation of Preference Information

Disadvantages:

a) human effort is required during the whole search process;

These methods usually progress by either changing the weights in a weighted-sum approach, or by progressively reducing the search space. The decision-maker then has to determine a relaxation to some objectives in order to achieve improvements in others. The relaxed objectives are moved from the objective function and added as constraints to limit the solutions space.
MOO: Formulating the Objective

**Posteriori Articulation of Preference Information**

- Search the solution space for a set of Pareto optimal solutions and present all of them to the user.
- The big advantages with these types of methods are that the solutions are independent from the users preferences.
- The analysis has only to be performed once, as the Pareto set would not change as long as the problem description remains unchanged.
- **However**, some of these methods suffer from a large computational burden and another disadvantage might be that the decision-maker has too many solutions to choose from.
Pareto-based Techniques

- Evolutionary Algorithms seem particularly suitable to solve multi-objective optimization problems, because they deal simultaneously with a set of possible solutions (the so-called population).
- Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (they can easily deal with discontinuous or concave Pareto fronts), whereas these two issues are a real concern for traditional mathematical programming techniques.
Evolutionary MOO

Introduction

Optimization
- Intro & OR
- Comp. Intell.
- MOP

Final Remarks

MDAO
- Architectures
- Re-Entry Case
- Med. Devices
- Strath. Hemo
Evolutionary MOO

Introduction
Optimization
Intro & OR
Comp. Intell.
MOP
Final Remarks
MDAO
Architectures
Re-Entry Case
Med. Devices
Strath. Hemo
Evolutionary MOO

Introduction

Optimization
- Intro & OR
- Comp. Intell.
- MOP

Final Remarks

MDAO
- Architectures
- Re-Entry Case
- Med. Devices
- Strath. Hemo
Pareto-based Techniques

- Some approaches use the dominance rank, i.e., the number of individuals by which an individual is dominated, to determine the fitness values.
- Others make use of the dominance depth, where the population is divided into several fronts and the depth quantifies the front an individual belongs to.
- Alternatively, also the dominance count, i.e., the number of individuals dominated by a certain individual, can be taken into account.
- **Niching // Crowding techniques**
Evolutionary MOO

MOGA: the rank of an individual corresponds to the number of solutions in the current population by which it is dominated (dominance rank).

![MOGA Diagram](image)
Evolutionary MOO

NSGA: dominance depth

![Diagram showing NSGA results with dominance depth](image-url)
OPTIMIZATION GOOD PRACTICE
Solving Approaches

An appropriate application of the optimisation algorithms involves both recognizing what kind of system the user is dealing with and knowing the right algorithm to apply, but making these decisions is not easy.

First, we can distinguish three kind of systems and problem formulations:

1. **simple system and simple problem formulation**: when model functions and problem functions (objectives and constraints) are identifiable being linear or convex;

2. **complex system and complex formulation**: when model functions and problem functions are identifiable being non-regular, non-linear and with tightly coupled variables;

3. **unknown system and unknown problem**: when model functions and problem formulations are not identifiable.
Solving Approaches

- If the considered problem belongs to the case 1), since most of traditional techniques have been developed to handle these problems, the user can make a choice among several algorithms, which usually give excellent results.

- If the problem belongs to the case 2), the user could apply one of the above mentioned techniques, but bad results will be obtained almost always (at the best with only local optima), or he should apply some technique specifically developed to solve that problem (if there is one and the user knows that, of course).

- If the problem belongs to the case 3), the user can only try with every algorithm s/he knows, with the hope someone works (from a point of view, in this case we can speak of a complex problem as well, where complexity is due to the ignorance).
• If the considered problem belongs to the case 1), since most of traditional techniques have been developed to handle these problems, the user can make a choice among several algorithms, which usually give excellent results.

• If the problem belongs to the case 2), the user could apply one of the above mentioned techniques, but bad results will be obtained almost always (at the best with only local optima), or he should apply some technique specifically developed to solve that problem (if there is one and the user knows that, of course).

• If the problem belongs to the case 3), the user can only try with every algorithm s/he knows, with the hope someone works (from a point of view, in this case we can speak of a complex problem as well, where complexity is due to the ignorance).
Solving Approaches

- If the considered problem belongs to the case 1), since most of traditional techniques have been developed to handle these problems, the user can make a choice among several algorithms, which usually give excellent results.

- If the problem belongs to the case 2), the user could apply one of the above mentioned techniques, but bad results will be obtained almost always (at the best with only local optima), or he should apply some technique specifically developed to solve that problem (if there is one and the user knows that, of course).

- If the problem belongs to the case 3), the user can only try with every algorithm s/he knows, with the hope someone works (from a point of view, in this case we can speak of a complex problem as well, where complexity is due to the ignorance).
No-Free-Lunch (NFL) Theorems

• These theorems claim that “for any algorithm, any high performance over one class of problems is offset by performance over another class”, meaning that

• if an algorithm does particularly well on average for one class of problems then it must do worse on average over the remaining problems.

• As a consequence, any statement reporting performance of an algorithm outperform performance of another must be carefully qualified in terms of the set of problems under consideration.
MULTIDISCIPLINARY DESIGN OPTIMIZATION
Key technologies for MDO

**Environment**
- HPC facilities
- Efficient models
- Meta-modelling
- Decomposition/Aggregation
- Collaboration
- Concurrency

**Optimisation**
- Deterministic/Stochastic
- Multi-objective
- Constraints

**Uncertainty**
- Structure representation
- Quantification

**MDO approaches**
- Decision making
- Data mining

**Introduction**
- Intro & OR
- Comp. Intell.
- MOP
- Final Remarks

**MDAO**
- Architectures
- Re-Entry Case
- Med. Devices
- Strath. Hemo
Multi Disciplinary Design Optimisation

“Methodology for the design of complex engineering systems and subsystems that coherently exploits the synergism of mutually interacting phenomena” (Giensing & Barthelemy, “A Summary of Industry MDO Applications and Needs”, 1998)

“A field of engineering/mathematics that allows the modification of an existing design, including multidiscipline interactions, to arrive at a better one” (Alonso, workshop presentation, 2010)

**MDO is “just a tool” to help people make better decisions**
MDO: Advantages and difficulties

Advantages

- From sequential and partial optimisation to an integrated, more effective and efficient process
  - Sequential optimization may not lead to the true optimum and it is slow

![Image](Image by MIT OpenCourseWare)
MDO: Advantages and difficulties

**Advantages**

- From sequential and partial optimisation to an integrated, more effective and efficient process
  - Sequential optimization may not lead to the **true** optimum and it is slow

**Difficulties:**

- Interfacing (communication and “translation”)
- Computational Cost and Complexity
- Timing and scheduling
MDO Approaches

- When a single designer or team is able to develop and apply design tools across all disciplines, difficulties in communication and organization are minimized.

- As design problems become more complex, the role of decentralized disciplinary specialists increases and it becomes more difficult for a central group to manage the process.

- During the past 30 years, design has moved from near serial or a loosely coupled multidisciplinary approach to newer more integrated ones, having recognized that a serial or partial approach is slow and brings only suboptimal solutions, which in turn means long design times and high costs.
MDO Approaches

- Formal optimization procedures and tools were initially introduced for structural problems, then later extended to aero-structures and, progressively, to problems involving multiple disciplines, to respond to the need of a more holistic approach.

- MDO emerged as a new discipline providing a set of methods and tools to help engineers in the design of system for which the whole is greater than the sum of the parts.
• The main motivation for using MDO is that the performance of a multidisciplinary system is driven not only by the performance of the individual disciplines but also by their interactions.

• Considering these interactions in an optimization problem generally requires a sound mathematical formulation.

• By solving the MDO problem early in the design process and taking advantage of advanced computational analysis tools, designers can simultaneously improve the design and reduce the time and cost of the design cycle.
MDO Architecture/Approaches

• How to organize the discipline-analysis models, approximation models (if any), and optimization software in concert with the problem formulation so that an optimal design is achieved.

• Such a combination of problem formulation and organizational strategy is referred to as an MDO architecture.

• The MDO architecture defines both how the different models are coupled and how the overall optimization problem is solved.

MDO Architecture/Approaches

- The architecture can be either
  - monolithic or
  - distributed.
- In a monolithic approach, a single optimization problem is solved.
- In a distributed approach, the same problem is partitioned into multiple sub-problems containing smaller subsets of the variables and constraints.

MDO Terminology

• A design variable is a quantity in the MDO problem that is always under the explicit control of an optimizer.

• In traditional engineering design, the values of these variables are selected explicitly by the designer or design team.

• Design variables may be
  – *local*, i.e., pertain to a single discipline, or
  – *shared* by multiple disciplines.

MDO Terminology

- **Discipline analysis**: a simulation that models the behaviour of one aspect of a multidisciplinary system.
- Most disciplines are required to exchange *coupling variables* to model the interactions of the whole system.

- In many formulations, copies of the coupling variables must be made to allow discipline analyses to run independently and in parallel. These copies, which function as design variables in the problem formulation, are sometimes called *target variables*.

MDO Approaches

- Several MDO methods have been developed to handle the complexity of the integration of multiple disciplines.
- Some examples are:
  - all-at-once (AAO) [monolithic],
  - multiple discipline feasible (MDF) [monolithic],
  - individual discipline feasible (IDF) [monolithic],
  - simultaneous analysis and design (SAND) [monolithic],
  - concurrent subspace optimization (CSSO) [distributed], and
  - collaborative optimization (CO) [distributed].

MDO Approaches

- In the AAO method, there is a single loop in which the feasibility of each discipline is not maintained at each optimization step, and the optimizer handles both the minimization/maximization of the cost function and the feasibility of each discipline.
- This form of the design-optimization problem includes all coupling variables, coupling variable copies, state variables, consistency constraints, and residuals of the governing equations directly in the problem statement.
MDO Approaches

\[ \text{minimize } f_0(x, y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i) \]

with respect to \( x, \hat{y}, y, \bar{y} \)

subject to \( c_0(x, y) \geq 0 \)

\[ c_i(x_0, x_i, y_i) \geq 0 \quad \text{for } i = 1, \ldots, N \]

\[ c^c_i = \hat{y}_i - y_i = 0 \quad \text{for } i = 1, \ldots, N \]

\[ R_i(x_0, x_i, \hat{y}_{j \neq i}, \bar{y}_i, y_i) = 0 \quad \text{for } i = 1, \ldots, N \]

\( \bar{y} \) = vector of state variables (variables used inside only one discipline analysis)

Subscripts

\( i \) = functions or variables that apply only to discipline \( i \)

\( 0 \) = functions or variables that are shared by more than one discipline

Superscripts/Oversets

\( * \) = functions or variables at their optimal value

\( \sim \) = approximations of a given function or vector of functions

\( ^\wedge \) = independent copies of variables distributed to other disciplines

- **Introduction**
- **Optimization**
  - Intro & OR
  - Comp. Intell.
  - MOP
- **Final Remarks**

**MDAO**

- Architectures
- Re-Entry Case
- Med. Devices
- Strath. Hemo
MDO Approaches

- **Monolithic** MDO architectures: those that form and solve a single optimization problem.

- **Distributed** MDO architectures: decompose the optimization problem into a set of smaller optimization problems, or sub-problems, that have the same solution when reassembled.

- The motivation for decomposition methods was to exploit the structure of the problem to reduce solution time.
MDO Approaches

• While decomposition methods do exist for nonlinear problems, the problem structure is NOT the primary motivation for their development.

• Primary motivation comes from the structure of the engineering-design environment:
  – Typical industrial practice involves breaking up the design of a large system and distributing it among specific engineering groups.
MDO Approaches

- Furthermore, design groups typically like to retain control of their own design procedures and make use of in-house expertise, rather than simply passing on the discipline-analysis results to a central design authority.

- Decomposition through distributed architectures allows individual design groups to work in isolation, controlling their own sets of design variables, while periodically receiving updated information from other groups to improve their aspect of the overall design.
MDO Approaches: Timing

- The structure of discipline design groups working in isolation has a profound effect on the **timing** of each discipline-analysis evaluation.

- In a monolithic architecture, all discipline-analysis programs are run exactly the same number of times, based on requests from the optimizer or MDA program. (In the context of parallel computing, this approach can be thought of as a **synchronous** algorithm)

- Where some analyses or optimizations are much more expensive than others, such as **multi-fidelity** optimization, the performance suffers because the processors performing the inexpensive analyses and optimizations experience long periods of inactivity while waiting for updates from other processors.
MDO Approaches

- By decomposing the optimization problem, we can balance the processor workloads.
- Those disciplines with less demanding optimizations may also be allowed to make more progress before updating nonlocal information. (the design process occurs not only in parallel but also *asynchronously*).
- Overall, this may result in more computational effort, but the intrinsically parallel nature of the architecture allows much of the work to proceed concurrently, *reducing the wall-clock time*. 
• Besides the integration of multiple disciplines, another critical aspect of MDO is the *fidelity* of the simulation models for each discipline.

• To keep the computational cost within reasonable terms, only reduced or low-fidelity models could be used during the optimization process, leaving *high-fidelity* simulations to further analyses.

• There is, however, a clear need to integrate high-fidelity simulations within the optimization process from the very beginning of the design.
MDO: Model Fidelity

- The use of high-fidelity models would allow the capture of important phenomena that low fidelity models might not incorporate, and would improve the design process by providing sufficiently accurate solutions to make correct decisions without the need for further verification analysis.
Several solutions have been proposed and implemented, all which have in common the combined use of low-fidelity models or high-fidelity models with some kind of surrogate models.

Some approaches use only data-fitted approximated models (or surrogates), such as response surfaces, Kriging, and Artificial Neural Networks (ANNs), which are improved during the design process.

Their response eventually converges to the high-fidelity model response in the proximity of the optimal solution(s).
MDO: Model Fidelity

- Several solutions have been proposed and implemented, all which have in common the combined use of low-fidelity models or high-fidelity models with some kind of surrogates.
- Some approaches use only data-fitted approximated models (or surrogates), such as response surfaces, Kriging, and Artificial Neural Networks (ANNs), which are improved during the design process.
- Their response eventually converges to the high-fidelity model response in the proximity of the optimal solution(s).
Several solutions have been proposed and implemented, all which have in common the combined use of low-fidelity models or high-fidelity models with some kind of surrogate models.

Some approaches use only data-fitted approximated models (or surrogates), such as response surfaces, Kriging, and Artificial Neural Networks (ANNs), which are improved during the design process.

Their response eventually converges to the high-fidelity model response in the proximity of the optimal solution(s).
MDO: Model Fidelity

- Several solutions have been proposed and implemented, all of which combine the use of low-fidelity models or high-fidelity models with some kind of surrogate models.
- Some approaches use only data-fitted approximated models (or surrogates), such as response surfaces, Kriging, and Artificial Neural Networks (ANNs), which are improved during the design process.
- Their response eventually converges to the high-fidelity model response in the proximity of the optimal solution(s).
Other approaches use a hierarchical strategy, also called multi-fidelity, where physics-based models of increasing fidelity are employed during the design process, with only the last steps of the optimization being performed using the highest fidelity model.

A third type of approach combines data-fitted approximating models and multi-fidelity modelling, where the surrogate acts both as knowledge repository and information bridge between models with different fidelity.
Introduction

Optimization

Intro & OR
Comp. Intell.
MOP
Final Remarks

MDO: Model Fidelity

- Note that although it has been demonstrated that the use of approximated models is a valid approach to reduce the computational load, the burden of dimensionality can limit its applicability to low dimensional problems, and much effort is currently being mode to develop and implement new scalable methods both for design optimization and uncertainty propagation.
Robust Optimization

- Optimization of the design under uncertainties
  - Find the most reliable solution under problem uncertainties
  - Uncertainties in model parameters or in the design variables
  - Optimize towards maximization of the pump life cycle and performance

- Reliable models that are able to reproduce close to reality results
- Availability of necessary resources (computational and commercial licenses)

"Design Optimization" I. MacQueen, A. Riccardi, E. Minisci, A. Iannetti, M. Stickland
Integration Levels

Response of the system only function of shape/structure design parameters $\mathbf{d}$

Response of the system function of shape/structure design parameters $\mathbf{d}$ and control law parameters $\alpha$

Uncertain response of the system function of shape/structure design parameters $\mathbf{d}$ and control law parameters $\alpha$
Integration of system design and optimal control into a single optimisation process

**Idea**, hybrid optimisation technique integrating
- multi-objective (multi-constrained) stochastic/evolutionary algorithm,
- optimal control solvers,
- uncertainty quantification
Integrated Framework

**Introduction**

- Optimization
  - Intro & OR
  - Comp. Intell.
  - MOP
  - Final Remarks

**MDAO**

- Architectures
  - Re-Entry Case
  - Med. Devices
  - Strath. Hemo

**Integrated Framework**

- Multi-Fidelity Evolution Control
- Multi/Single-objective Optimisation
- Uncertainty based MDO
- Optimal Control

**Uncertainty**

- Aleatory
- Epistemic
- Probabilities
- Evidence

**IDEA**

- MO-EDA
- Multiagent memetic

**Direct MShooting**

- PS Methods
- Finite Elements

**PS**

- Methods

29-June-15

Third talk at University of Cagliari
USV: Geometry and Shape Model

The planform and the upper surfaces of the vehicle are parameterized by the length $l$, the width, $w$, a power law exponent $n$, the vehicle center line wedge angle, $\theta$, and the oblique shockwave inclination angle $\beta$. 
Two different models are used to predict the aerodynamic characteristics of the vehicle.

- A simplified analytical model, which is here applied to the actual rounded-edge vehicle, although it was originally developed to predict the aerodynamics of the original sharp-edge shape of the waverider configuration.

- A computational fluid dynamic (CFD) model based on a finite volume integration of Reynolds Averaged Navier-Stokes equations (RANS).

- No thermo chemistry is included in the model.

- Unmodelled components are introduced as uncertainties in the aerodynamic parameters.

- Design parameters are chosen to minimise the impact of the uncertain quantities.
The surrogate model is an Artificial Neural Network (ANN) approximators. A generic Multi Layer Perceptron (MLP) ANN with one hidden layer was used.

The training process is based on a Bayesian regularization back-propagation, which limits any overfitting problem.

The idea is that the computational cost of the initial training and the online update are negligible if compared to a call to the high-fidelity model.

The inputs to the ANN approximator are the 5 geometric parameters, the angle of attack, the speed, and the altitude.

The outputs are the coefficient of lift, $C_L$ and drag, $C_D$.

The networks are trained to reach a mean squared error of 5% on the normalized training output.
Introduction

Optimization

Intro & OR
Comp. Intell.
MOP
Final Remarks

MDAO

Architectures
Re-Entry Case
Med. Devices
Strath. Hemo

USV System Models: TPS and Thermal Model

• The thermal protection system (TPS) is assumed to be made of Zirconium Diboride (ZrB$_2$) UHTC

<table>
<thead>
<tr>
<th>Properties</th>
<th>unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>density</td>
<td>6000 kg/m$^3$</td>
</tr>
<tr>
<td>specific heat</td>
<td>628 J Kg$^{-1}$ K$^{-1}$</td>
</tr>
<tr>
<td>thermal conductivity</td>
<td>66 W m$^{-1}$ K$^{-1}$</td>
</tr>
<tr>
<td>emissivity</td>
<td>0.8</td>
</tr>
</tbody>
</table>

• The whole nose cone is made of UHTC with thickness $L_{TPS}$. The rest of the vehicle is covered with a thin shell with a constant thickness of 0.002m
• The convective heat flux

\[ \dot{q}_{\text{conv}} = K_e \sqrt{\frac{\rho_{\infty}}{R_n}} V_n^3 \]

where \( K_e = 1.742 \times 10^{-4} \) (the heat flux is in W/m²)

• The internal temperature (\( T_{\text{int}} \)) is computed by solving the following one-dimensional heat equation:

\[ \frac{\partial^2 T}{\partial x^2} = \frac{c \rho_{T_{\text{PS}}}}{k} \frac{\partial T}{\partial t} \]

with boundary conditions:

\[ \dot{q}_{\text{conv}} - \varepsilon \sigma T_w^4 + k \frac{dT}{dx} \quad \text{at} \quad x = 0 \]

\[ k \frac{dT}{dx} = \varepsilon \sigma T_{\text{int}}^4 \quad x = L_{T_{\text{PS}}} \]

where \( c \) is the heat capacity, \( \rho_{T_{\text{PS}}} \) is the density of the TPS material, \( k \) is the thermal conductivity, \( \varepsilon \) is the material emissivity, and \( \sigma \) is the Stephen-Bolzmann’s constant.
USV System Models: Mass

• The total mass of the USV is made of the structural mass \( m_{st} \), the mass of the TPS \( m_{TPS} \) and the mass of the payload (avionics and power system) \( m_{pl} \).

\[
m = m_{TPS} + m_{st} + m_{pl}
\]

• The mass of the TPS is made of the mass of the nose \( m_{nose} \) plus the mass of the thin skin covering the rest of the vehicle \( m_{skin} \).

• The structure of the vehicle is supposed to be made of titanium, with a density of 4000 kg/m\(^3\).

• The structural mass \( m_{st} \), can be obtained as

\[
m_{st} = \rho_{body}(2S_p + S_b)d_{body}
\]

where \( d_{body} = 0.004 \) m is the thickness of the structure of the vehicle, seen as a shell.
USV System Models: Dynamic Equations and Optimal Control Sub-problem

- The vehicle is considered to be a point mass, whose motion is governed by the following set of dynamic equations

\[
\begin{align*}
\dot{r} &= v \sin \theta_p \\
\dot{\lambda} &= \left( \frac{v \cos \theta_p \cos \xi}{r} \right) \\
\dot{\phi} &= \left( \frac{v \cos \theta_p \sin \xi}{r} \right) \\
\dot{v} &= -\frac{D(\alpha)}{m} - g \sin \theta_p \\
\dot{\theta}_p &= \frac{L(\alpha)}{m v} \cos \gamma_v - \left( \frac{g}{v} - \frac{\dot{v}}{r} \right) \cos \theta_p \\
\dot{\xi} &= \frac{L(\alpha)}{m v \cos \theta} \sin \gamma_v - \frac{v}{r} \cos \theta_p \cos \xi \tan \phi
\end{align*}
\]

where \( r \) is the norm of the position vector with respect to the center of the planet, \( \lambda \) is the longitude, \( \phi \) the latitude, \( v \) the magnitude of the velocity, \( \theta_p \) is the flight path angle, \( \xi \) is the heading angle (azimuth of the velocity).

No out of plane maneuvers are considered, thus \( \gamma_v \) is kept equal to zero during the whole trajectory.
USV System Models: Dynamic Equations and Optimal Control Sub-problem

• The angle of attack $\alpha$ is the control variable therefore for each geometry the following optimal control sub-problem needs to be solved:

$$\min_{\alpha} \max_{t} q$$

subject to dynamic equations and terminal conditions

$$\begin{align*}
  r(t = 0) &= r_0 \\
  \lambda(t = 0) &= \lambda_0 \\
  \phi(t = 0) &= \phi_0 \\
  v(t = 0) &= v_0 \\
  \theta_p(t = 0) &= \theta_0 \\
  \xi(t = 0) &= \xi_0 \\
  r(t = t_f) &\leq r_f \\
  r(t = t_f) &\geq r_{min}
\end{align*}$$

The re-entry time is free and no other terminal conditions are imposed as there is no specific requirement on the landing point.
Case Study: Nano-USV

• A population of individuals evolves multiple system design solutions in parallel using a population based optimisation algorithm (Evolutionary Algorithm, EA)
  – for each system design solution, an optimal control profile is generated by solving an optimal control problem;
  – uncertainties on models (such as aerodynamic forces) and vehicle shape characteristics are integrated in the design process
  – EA searches for geometries, subject to constraints, that optimise and reveal statistical characteristics of objectives.

• Meta-modelling techniques are used to approximate expensive models.

• Meta-models are trained and updated by means of a multi-fidelity evolution control approach.
Robust multidisciplinary design

The mean, $E_q$ and $E_T$, and the variance, $\sigma_q^2$ and $\sigma_T^2$, of all the computed maximum heat flux and maximum internal temperatures were used as performance indexes. Based on this definition of the performance indexes, the robust design optimization under uncertainties can be formulated as follows:

$$\min_{d \in D} [E_q, E_T, \sigma_q^2, \sigma_T^2]$$

subject to the following constraints on the variance:

$$\sigma_q^2 \leq \bar{\sigma}_q^2; \sigma_T^2 \leq \bar{\sigma}_T^2$$

the design vector $d$ is defined as $d = [l, w, n, \theta, R_n, L_{TPS}]$
Robust multidisciplinary design

The MOO problem was solved with a particular type of evolutionary algorithm which belongs to the sub-class of Estimation of Distribution Algorithms (EDAs).

The specific EDA employed in this work is a multi-objective optimization algorithm for continuous problems that uses the Parzen method to build a probabilistic representation of Pareto optimal solutions, with multivariate dependencies among variables (Multi-Objective Parzen based Estimation of Distribution, MOPED).

Non-dominated sorting and crowding operators are used to classify promising solutions in the objective space, while new individuals are obtained by sampling from the Parzen model.
Robust multidisciplinary design

Transcription with a Gauss pseudospectral method and with Finite Elements in Time on spectral basis. (The two approaches gave similar results therefore it was decided to omit from this paper the comparison between the two approaches on this particular problem).

In both cases, the trajectory is decomposed in $n_e$ elements, each of which have $n_p$ collocation points.

After transcription, the optimal control problem becomes the following general nonlinear programming problem:

$$\min_{\alpha_s} \max_{t_s} \max_{\dot{q}_s}$$

subject to the nonlinear:

$$C(r_s, \lambda_s, v_s, \xi_s, \theta_s, \alpha_s, t_s) = 0$$

algebraic constraints.

$$\begin{align*}
  r(t = 0) &= r_0 \\
  \lambda(t = 0) &= \lambda_0 \\
  \phi(t = 0) &= \phi_0 \\
  v(t = 0) &= v_0 \\
  \theta_p(t = 0) &= \theta_0 \\
  \xi(t = 0) &= \xi_0 \\
  r(t = t_f) &\leq r_f \\
  r(t = t_f) &\geq r_{min}
\end{align*}$$
Robust multidisciplinary design

• The MOO optimization algorithm MOPED is integrated with an external procedure that monitors the status of the approximated models.

At the end of each iteration (generation), the external procedure checks if an updated version of the approximated model is ready and available.
Robust multidisciplinary design

• The MOO optimization algorithm MOPED is integrated with an external procedure that monitors the status of the approximated models.

In an asynchronous way, an additional external procedure manages the training and updating of the approximated model.

- Random (uniform) population of \( n_{\text{ind}} \) individual
- Evaluation of the population & Classification
- Parzen estimation of distribution & Sampling
- Class., Fitness, selection of the best \( n_{\text{ind}} \) ind.
- Convergence criterion satisfied?
- No
- recompute and reclassify the population
- Yes
- updated models?
- NO
- reclassify the population
- YES
- Check solutions and generations
- Update Database
- Initialize meta-modeling
- Initial ANN
- Random (uniform) population of \( n_{\text{ind}} \) individual
Case Study: Nano-USV

- Optimal design of a nano autonomous re-entry vehicle.
- Minimum heat flux and maximum cruise distance
- Vehicle as small as possible
Case Study: (Re)-Entry Capsule

- **Two universities**
  - University of Strathclyde
  - Politecnico di Torino

- **Three enterprises**
  - Thales Alenia Space
  - Optimad
  - Exemplar

- **Regional project**
  - STEPS
Case Study: (Re)-Entry Capsule
Case Study: Wind Turbines

• Multi-disciplinary robust design of variable speed wind turbines
Current and Future Capabilities

- Further development of single components
  - Development and validation of specific and **realistic models**;
  - Modelling of **realistic uncertainty structures**, which should be integrated into the design process, and the verification and validation of obtained uncertainty models;

- Integration of all components and application to **real case studies**, based on historical data;

- Development of **user interfaces**, which should be intuitive and really centred on human characteristics.
Future Capabilities

- Implementation of the tools in the **Concurrent Design Facility** in the Strathclyde TIC building
- Extension to collaborative design
- CDF and tools available onsite or remotely
Summary

• Introduction

• Optimization
  – Introduction & OR
  – Single-Objective Problems
  – Computational Intelligence
  – Multi-Objective Problems
  – Final Remarks

• Multidisciplinary Design Analysis and Optimization
  – Architectures
  – Re-Entry Test Case
  – MDAO of Medical Devices
  – Hemodynamics at Strathclyde
THANKS!

Edmondo Minisci (edmondo.minisci@strath.ac.uk)
We will organise EUROGEN 2015

More info at www.strath.ac.uk/eurogen2015
Times Higher Education
UK Entrepreneurial University of the Year