Multi objective optimisation examples: design of a laminar airfoil and of a composite rectangular wing

V.Pediroda, M.Poian, Carlo Poloni
Dipartimento di Energetica
Università degli Studi di Trieste
Via Valerio 10, 34100 Trieste, ITALY
e-mail: poloni@univ.trieste.it

Introduction

As industrial optimisation moves from being applied to particular systems or structures towards addressing the total scope of a design, there is a need for capability to express and deal with trade-offs of conflicting objectives which measure a design's merit in a number of business senses. This leads to a need for:

1. a structured set of business metrics;
2. an organisation which is adapted to address shared responsibilities for design choices across disciplines and companies where necessary;
3. design evaluation tools to address all, not just some, of these metrics;
4. supporting technology to provide:
   - ✓ systems integration in a wide area distributed environment,
   - ✓ user-oriented design analysis facilities,
   - ✓ appropriate optimisation tools, and
   - ✓ decision support tools to assist in understanding the compromises involved.

Each design organisation has to address the first three of these needs in ways which suit its business situation [1],[2],[3]. This is true even if the design problem is purely related to aerodynamics but has to be linked to real application.
To solve and optimisation problem three basic ingredients must exist:

- optimisation algorithms
- reliable simulation software
- appropriate Information Technology infrastructure

A design is assumed to be a definition of a product or process which:

- is complete enough to allow the significant measures of its business effectiveness to be evaluated (preferably objectively via a defined evaluation process)
- is one candidate from a consistently defined class of designs, which allows 'cause and effects to he established between a design change and the design's business effectiveness and is likely to be realisable in practice, with current knowledge of likely constraints.

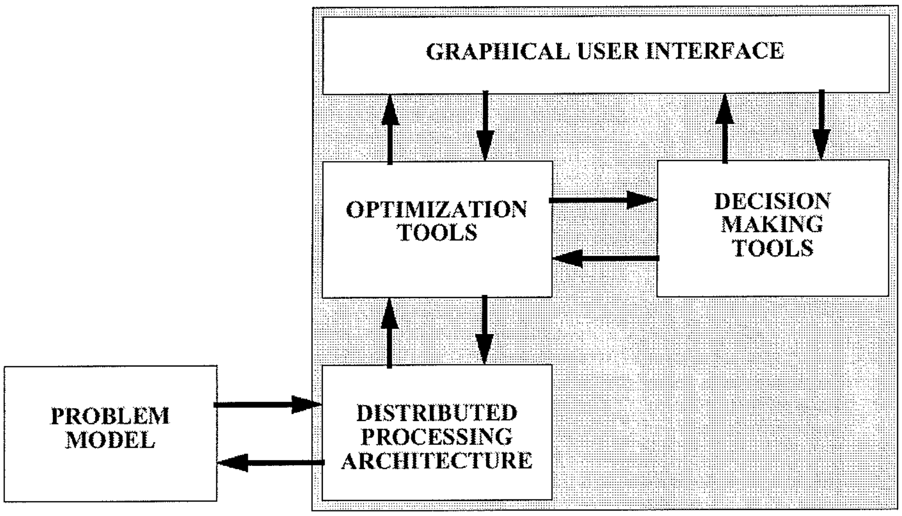
No single metric will suffice to characterise all significant product qualities. For example, it is usual to aim to achieve a balance between maximum performance and minimum cost or to

achieve a balance between different operating conditions. Therefore, at product level, optimisations are trade-offs between several objectives.
In the examples illustrated in this paper FRONTIER System [4] has been used in order to cover the aspects mentioned above.

Frontier system

In software terms the FRONTIER system is a general purpouse design tool which enables the designer to search for optimal designs over a number of criteria (e.g. cost, efficiency,...) within a given design space. The designs generated during optimisation form the trade-off surface (Pareto Frontier) which is then passed into a selection tool which is used in selecting a single design which best suits the designer's aspirations.

The system is made up of a number of distinct parts which are integrated to form the complete FRONTIER system (Figure 1):



1. GUI - Graphical User Interface, includes visualisation and setup tools.
2. Optimization Tools - Several methods used to perform single/multiple objective optimisation.
3. Distributed Processing Architecture - Manages processes on several machines.
4. Decision Making Tools - Aids designer in selection of 'best' design.
5. Problem Models - Provided by the user to perform search on.

The user interface uses a browser type front end. This allows the system to be implemented using a combination of HTML (Used in web pages) and Java/JavaScript. Such an architecture was chosen because of its portability across systems and as such FRONTIER had been tested successfully on NT, Unix, Linux, SGI, and HP operating systems.
The system can use both traditional optimisers (Simplex, BFGS, SQP) Multi Objective Genetic Algorithm (MOGA) specifically designed for computationally intensive applications [5]. Optionally the user can compile his own optimisation library with the available application interfaces.