Multi-objective optimization and decision making process in engineering design

Companies need to optimize their products and processes daily, hence optimization plays a significant role in today's design cycle. Problems related to one or more than one objective, originate in several disciplines; their solution has been a challenge to engineers for a long time. Typically using a single optimization technology is not sufficient to deal with real-life problems. Therefore, engineers are frequently asked to solve problems with several conflicting objective functions. The definition of Multi-disciplinary optimization given by the Multi-Disciplinary Optimization Technical Committee of the American Institute of Aeronautics and Astronautics (AIAA) is self-explaining. The definition states: "Optimal design of complex engineering systems which requires analysis that accounts for interactions amongst the disciplines (or parts of the system) and which seeks to synergistically exploit these interactions".

Multi-objective optimization

It is difficult to explain optimization techniques in a few words; there are plenty of books describing different methods and approaches.

Roughly speaking, to optimize means selecting the best available option from a wide range of possible choices. This can be a complex task as, potentially, a huge number of options should be tested when using a brute force approach. There are several sources of complexity, such as the computational difficulties in modeling the physics, the potentially high number of free variables, or a high number of objectives and constraints. Moreover, the coupling between disciplines can be challenging, involving several complicating factors, such as the limitation on the computational resources, restrictions connected with the algorithms' capabilities and even a lack of communication between different departments. Often, engineers are so committed to a single position that they lack an overall picture of the optimization problem. Anyhow, the design analysis can be decomposed into different levels, where each "sub-structure" can be approached and optimized in a similar way. By using design automation procedures, the entire design process or the specific sub-problems can be analyzed systematically, by means of:

- Design of Experiments
- Optimization Algorithms
- Decision-Making Procedures

In a previous article, we explained how the Design of Experiments tool can help in preparing and executing a given number of experiments in order to maximize knowledge acquisition. In this article, we focus on optimization algorithms and decision making procedures.

In order to help engineers and decision makers, old and new optimization techniques are studied and widely used in industries. Each optimization technique is qualified by its search strategy that implies the robustness and/or the accuracy of the method. The robustness of an optimization method is the ability to reach the absolute extreme

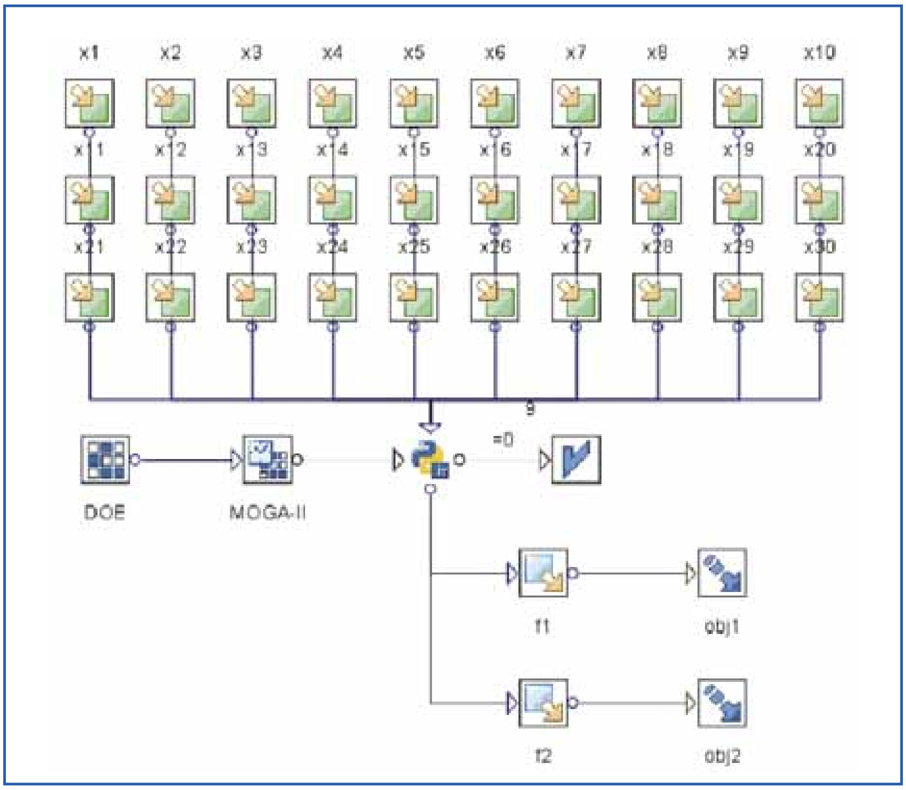

Figure 1: modeFRONTIER workflow describing a well-known ZDT1 multi-objective problem. This problem has 30 continuous input variables and two objectives. A complete description of this problem can be found in K. Deb's, "Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems", Evolutionary Computation, 7(3):205-230.

of the objective function even when starting far away from the final solution. On the contrary, the accuracy measures the capability of the optimization algorithm to get as close as possible to function extreme. There are hundreds or thousands of optimization methods in literature, each numerical method can solve a specific or more generic problem. Some methods are more appropriate for constrained optimization, others may be suitable for unconstrained continuous problems, or tailored for solving discrete problems. A lot of “classical” optimization methods exist; these methods can be used provided that certain mathematical conditions are satisfied. Thus, for example, linear programming efficiently solves problems where both the objective and the constraints are linear with respect to all the decision variables. Other specific numerical methods can be useful for solving quadratic programming, nonlinear problems, nonlinear least squares, nonlinear equations, multi-objective optimization, and binary integer programming. Unfortunately, real world applications often include one or more difficulties which make these methods inapplicable. Most of the time, objective functions are highly non-linear or even may not have an analytic expression in terms of the parameters.

The mathematical formulation of a general multi-objective optimization problem can be written as follows:

$$\min\left[f_1(x_1,\ldots,x_n), f_2(x_1,\ldots,x_n),\ldots, f_k(x_1,\ldots,x_n)\right]$$

$$\text{subject to}\begin{cases} g_i(\bar{x}) \geq 0 \\ g_l(\bar{x}) = 0 \\ \bar{x} \in S \end{cases}$$

When k>1 and the functions are in contrast, we speak about multi-objective optimization. (x1, …, xn) are the variables, the free parameters, i.e. the quantities that the designer can vary. It is by modifying these values that the search for an optimum is performed. The variables can be continuous or discrete. The problem may even contain a mixture of continuous and discrete variables. This does not pose any extra difficulties in setting the optimization problem up, however, it may slightly restrict the user's choice of search algorithms.

Adopting a model with a large number of input variables may appear to give more freedom of choice for the final design. However, the more dimensions the parameter space implies, the more work will be involved in searching the space for optimum designs. In practice, the work and hence the computational cost, snowball as the number of parameters increases.

(f1,…,fk) are the objective functions, the response parameters. These are the quantities that the designer wishes to maximize or minimize. For example, the designer can maximize the efficiency, the performance, or can minimize the cost, the weight. In multi-objective optimization problems, there are three possible situations: minimize all the objective functions, maximize all the objective functions, or minimize some and maximize others. For simplicity reasons, usually all the functions are converted to maximization or minimization form. Hence a maximization problem can always be transformed into a minimization problem with the following identity:

max fi(x) = -min(-fi(x))

Gi(x) are the constraints. Equality and inequality constraints are the quantities imposed on the project, i.e. restrictions and limits that the designer must meet due to the norms, or by the particular characteristics of the environment, functionalities, physical limitations, etc. These restrictions must be satisfied in order to be able to consider a certain solution as acceptable. All the constraints define a feasible region. The designer can impose some general constraints such as the maximum admissible stress, the maximum deformation, the minimum performance. Or the designer can even impose some special constraints on the variables such as the total volume, the range for the thickness, and so on.

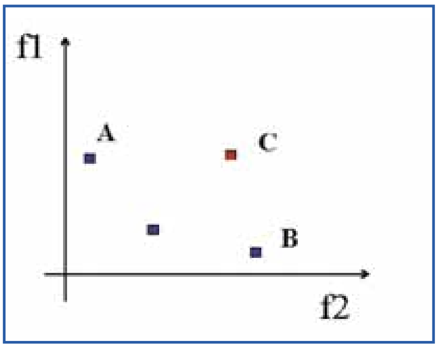

Figure 2: The concept of Pareto dominance. Point C is dominated by points A and B. A and B are better than C both for objective f1 and objective f2. A does not dominate B and B does not dominate A because A is the best point with respect to objective f2 and B is the best one with respect to objective f1. Hence, A and B are non-dominated and efficient solutions.

With a multi-objective problem, the notion of “optimum” changes as the aim is to find good compromises rather than a single solution. So, a multi-objective optimization does not produce a unique solution but a set of solutions. These solutions are called Pareto solutions, the set of solutions can be called both “trade-off surface” or Pareto frontier.

In the Pareto frontier, none of the points are “dominated”. By definition we say that the design a dominates design b if:

[f1(a) <= f1(b) and f2(a) <= f2(b)
…and fk(a) <= fk(b)]
and [f1(a) < f1(b) or f2(a) < f2(b)
…or fk(a) < fk(b)]

Roughly speaking, we can say that in the Pareto frontier none of the components can be improved without deterioration of at least one of the other components. Figure 2 shows the concept of Pareto optimal dominance in a 2-dimensional space. Pareto optimal solutions are also known as non-inferior, non-dominated or efficient solutions. These solutions may have no clear relationship besides their membership in the Pareto optimal set. Moreover, it may be difficult or even impossible to find an analytical expression of the surface that contains all these efficient points.

modeFRONTIER is a very powerful tool for multi-objective optimization, and it includes the most widely used methods. It offers an easy-to-use graphic user interface for describing the problem as shown in Fig. 1. modeFRONTIER contains both “classic” and metaheuristics methods for single and multi-objective optimizations. Metaheuristics methods are a new type of methods that have been developed since 1980. These methods have the ability to solve even difficult optimization problems in the best way possible. This is an important group of methods that has significantly contributed to the renewal of multi-objective optimization. Before, multi-objective optimization problems were solved only by means of weighted functions with which the problem is transformed into a single objective problem using weights wi:

F(x) = w1*f1+w2*f2+…+wk*fk

This formulation seems to be very simple and easy to understand although it may seem like adding apples and oranges. And unfortunately, this simple formulation has several drawbacks. First of all, the weights are problem-dependent and must be empirically defined by the user. The user should even take care of normalization and this can be as complex as the global optimization because the range of variation of each function may be unknown. Last but not least, summing up functions even means summing up discontinuity. Until recently, considerations of computational expense forced users to use only this kind of weak approach. However, newer and more ambitious approaches such as the so-called metaheuristics methods aim to optimize several objectives simultaneously, thus generating various points in the Pareto set. The class of methods includes among others: simulated annealing, genetic algorithms, particle swarm, ant colonies, evolutionary strategies, tabu search. They have some characteristics in common, such as to be at least partly stochastic and not to require to compute derivatives. Moreover, they are inspired by analogies with physics, or with biology or with ethology. Unfortunately, they even share the same drawback which is, usually, the high computation time required for convergence. This should be considered as the price to be paid in order to have a robust method that has the ability to overcome the obstacle of local optima. This problem is partially solved by the parallelization: in recent years, several ways of parallelizing various metaheuristics have been proposed.

All these metaheuristics are not mutually exclusive. It is often impossible to predict with certainty the efficiency of a method when it is applied to a problem. This statement is confirmed by the well-known “no-free-lunch theorem” (NFLT) developed by D. Wolpert and W. Macready. The theorem states that "[...] all algorithms that search for an extremum of a cost function perform exactly the same, when averaged over all possible cost functions."

For this reason, modeFRONTIER includes a wide range of possible algorithms that can be selected for solving different problems. At present, the methods available in modeFRONTIER are:

- SIMPLEX
- Bounded-BFGS
- Levenberg-Marquardt
- Simulated Annealing
- Multi-Objective Genetic Algorithm (MOGA)
- Adaptive Range MOGA
- Multi-Objective Simulated annealing (MOSA)
- Non-dominated Sorting Genetic Algorithm
- Multi-objective Game Theory
- Fast MOGA
- Fast SIMPLEX
- Evolutionary Strategies Methodologies
- NLPQLP
- Normal Boundary Intersection (NBI)

Moreover, different algorithms can be even combined to obtain some hybrid methods. An hybrid method can try to exploit the specific advantages of different approaches by combining more than one together. For example, it is possible to combine the robustness of a genetic algorithm together with the accuracy of a gradient-based method, using the former for initial screening and the latter for refinements. Whenever possible, modeFRONTIER’s algorithms can be used in parallel, to run more than one evaluation at once and to take advantage of available queuing systems.

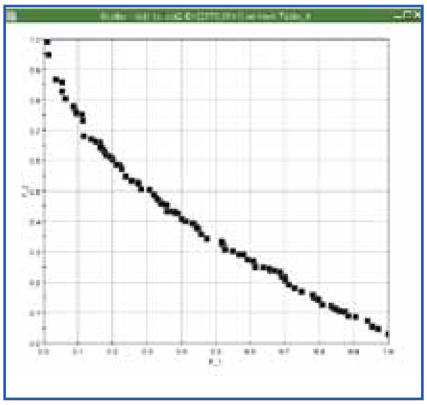

Figure 3: Pareto frontier of the ZDT1 obtained by using MOGA-II.

Multi-Criteria Decision Making (MCDM)

As shown in figure 3, modeFRONTIER can look for a complete set of non-dominated solutions. However and obviously, after having found some solutions of the multi-objective optimization problem, we are facing some difficulties: although many efficient solutions exist, only one or a reduced number of final solutions must be selected.

As it is impossible to order the full range of available designs (at once), all Pareto optimal solutions can be regarded as equally desirable in the mathematical

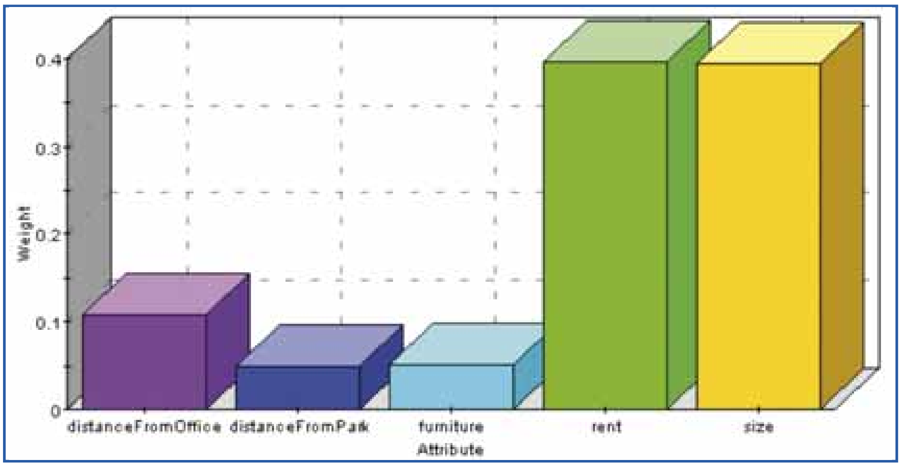

Figure 4: Decision Making example: Choosing a flat to rent on the basis of the following criteria: size, rent, distance from the office, distance from open spaces such as parks and quality of the furniture. The DM can express pair-wise relationships between attributes such as "rent is twice more important than distance from open spaces" and "distance from the office is twice more important than distance form the nearest park". This chart shows the automatic weights generated by modeFRONTIER according to DM choices, the final ranking depends on these weights.

sense. Hence, we need decision makers (DMs) to identify the most preferred one among the solutions. The decision maker is a person who is able to express preference information related to the conflicting objectives.

Ranking between alternatives is a common and difficult task, especially when several solutions are available or when many objectives or decision makers are involved. The decision makers choose one reasonable alternative from a limited set of available ones; design decisions usually reflect the competencies of each decision maker. When more than one decision attribute exists, making coherent choices can be a very difficult task.

Multi-Criteria Decision Making (MCDM) refers to the solving of decision problems involving multiple and conflicting goals, coming up with a final solution that represents a good compromise that is acceptable to the entire team. modeFRONTIER Multi-Criteria Decision Making allows the user to classify all the available alternatives through pair-wise comparisons on attributes and designs. Moreover, modeFRONTIER helps decision makers to verify the coherence of relationships. To be coherent, a set of relationships should be both rational and transitive. To be rational means that if the decision maker thinks that solution A is better than solution B, then solution B is worse than solution A. To be transitive means that if the decision maker says that A is better than B and B is better than C, then solution A should be always considered better than solution C.. So, we can say that the Multi Criteria Decision Making tool provided in modeFRONTIER assists the Decision Maker in finding the best solution from a set of reasonable alternatives. It allows the correct grouping of outputs into a single utility function that is coherent with the preferences expressed by the user and it does not have the same drawbacks of a weighted function.

The following algorithms are actually available in modeFRONTIER:

- Linear MCDM to be used when the number of decision variables is small;
- GA MCDM which does not perform an exact search but is more efficient than the previous method.
- Hurwicz used for uncertain decision problems. This criterion represents a compromise between the maximax and maximin criteria. The decision maker is neither optimistic nor pessimistic. With this criterion, the decision attributes are weighted by a coefficient that is a measure of the decision maker's optimism. For example, when the Hurwitz weight is equal to zero, the maximax criterion is used. With this criterion, the decision maker selects the design that represents the maximum of the best attribute. On the contrary, when the Hurwitz weight is equal to one, the reverse approach to the maximax criterion is used. The maximin criterion is based on the assumption that the decision maker is pessimistic about the future. With this criterion, the minimum value of the attributes for each designs are compared, and the design that produces the maximum of the minimum value must be chosen;
- Savage MADM used for the uncertain decision problems where both the decision states and their likelihoods are unknown. This algorithm examines the regret (i.e. losses) resulting when the value of the selected alternative is smaller than the optimized value. Then, the minimax criterion suggests that the decision maker should look at the maximum regret of each strategy selecting the one with the smallest value.

Conclusions

The website www.esteco.com contains several examples of how to use Multi-Objective Optimization and Decision Making Process in Engineering Design. Most of the examples are slanted towards applications in fluid dynamics. However, this only reflects some of the research interests of the original authors since multi-objective optimization problems and the coupling of these techniques with modeFRONTIER can be much more general.

Silvia Poles
ESTECO - Research Labs
scientific@esteco.com