12

Multiobjective Optimization Software

Silvia Poles[1], Mariana Vassileva[2], and Daisuke Sasaki[3]

[1] ESTECO - Research Labs, Via Giambellino, 7 35129 Padova, ITALY
silvia.poles@esteco.com
[2] Institute of Information Technologies, Bulgarian Academy of Sciences,
BULGARIA mvassileva@iinf.bas.bg
[3] CFD Laboratory, Department of Engineering, University of Cambridge,
Trumpington Street, Cambridge CB2 1PZ, UK ds432@eng.cam.ac.uk

Abstract. This chapter provides a description of multiobjective optimization software with a general overview of selected few available tools developed in the last decade. This chapter can be considered a revision of previous valid papers and chapters on nonlinear multiobjective optimization software such as the ones written by Weistroffer *et al.* (2005) and Miettinen (1999) that lists existing software packages up to the year 1999. More precisely, this chapter is focused on the tools and features that advisable multiobjective optimization software should contain.

12.1 Introduction

The main topic to be discussed in this chapter is available multiobjective optimization software. The main concern is devoted to software developed for nonlinear problems. Several questions may be raised when discussing multiobjective optimization software, but among the most recurring questions we may list the following:

- What do experts think about multiobjective optimization tools and what are the most important features good software should always possess?
- What is the current state-of-the-art of multiobjective optimization software?
- What are the advantages and gaps of all these optimization tools?

The description of an ideal software is very close to a complex integrated environment such as a "Process Integration and Design Optimization" (PIDO) or a "Problem Solving Environment" (PSE) (Gallopoulos *et al.*, 1991; Houstis *et al.*, 1997). PIDO and PSE are integrated computing environments which

Reviewed by: Oliver Bandte, Icosystem Cooperation, USA
Jyrki Wallenius, Helsinki School of Economics, Finland
Kaisa Miettinen, University of Jyväskylä, Finland

J. Branke et al. (Eds.): Multiobjective Optimization, LNCS 5252, pp. 329–348, 2008.

provide the users all the necessary tools for solving multiobjective optimization problems and for supporting decision making.

An ideal tool should have: an easy-to-use graphical user interface, a good set of optimization methods, a good tool for visualizing the results and choosing the final solutions. Moreover meta-modeling and validation of models are fundamental when dealing with time-consuming function evaluations. Last but not least, robustness and reliability of solutions are of primary importance for selecting the best design.

There are many attributes and characteristics that can be used to measure software quality as seen by end-users. Leaving out all the problems related to reliability, absence of bugs, extensibility and maintainability of each tool, we here refer to requirements that a decision maker may have for a multiobjective optimization software.

In the following sections a list of advisable program specifications is explained. Next, a list of software is described and their conformance to requirements and specifications is analyzed.

12.2 Software Features and Quality

12.2.1 Graphical User Interface

One of the most evident characteristics of a software is always a flexible, complete and easy-to-use graphical user interface (GUI). Even with multiobjective optimization tools, the GUI plays an important role. In this case, the GUI should give to the users of the software being it analysts or decision makers (e.g. engineers and managers) the ability to define and modify a problem, to define input, output, objectives and constraints. Moreover, the GUI should give to the decision makers the ability to choose optimization strategies, manage software and hardware resources, describe how the processes are synchronized and visualize and analyze results. Moreover, the GUI should be suitable for introducing decision maker's preferences in order to solve multiobjective decision making problems with an intelligent guidance.

For example, a multiobjective optimization problem can be described using graph-based formalisms as shown in Fig. 12.1. The figure describes a standard mechanical design problem, the design of a welded beam structure with the aim to minimize cost and displacement subject to constraints on shear.

12.2.2 Optimization Methods

Problems related to one or more than one conflicting objective functions, originate in several disciplines; their solution has been a challenge for a long time. Typically, using a single optimization technology is not sufficient to deal with real-life problems.

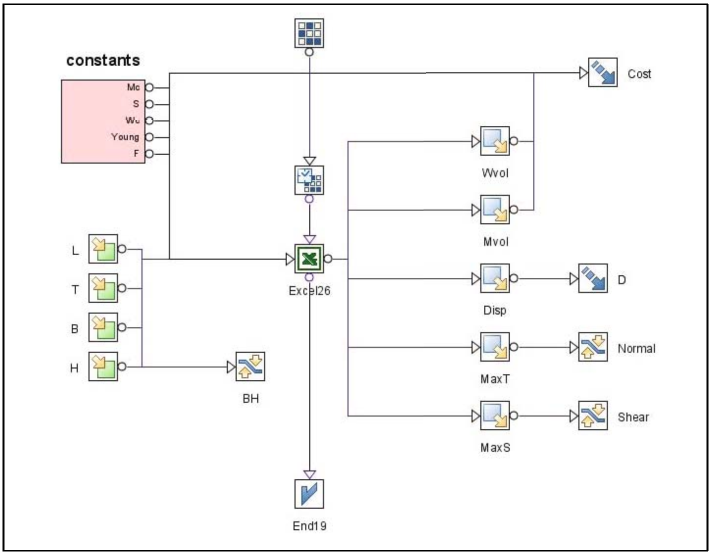


Fig. 12.1. An example of how a workflow can describe input, output, constraints and objectives of a multiobjective optimization problem.

In order to help engineers and decision makers, old and new multiobjective optimization techniques are studied in industries, project and portfolio management, military and governmental fields. The importance of managing more than one objective at once as opposed to just optimizing one outcome is well recognized, for example, in portfolio management. In fact, constructing a balanced bond portfolio must deal with uncertainty in the future price of bonds and several other aspects. Despite what is reported in (Kaliszewski, 2004), multiobjective optimization has recently started to gain attention within the engineering and scientific communities since many real world optimization problems in numerous disciplines and application areas, contain more than one outcome that should be optimized.

Each optimization technique is qualified by its search strategy that implies the robustness and/or the accuracy of the method. An indication of the robustness of an optimization method is the ability to reach the absolute extremes of the functions even when starting far away from the final solutions. On the contrary, the accuracy measures the capability of the optimization algorithm to get as close as possible to function extremes. There are hundreds or thousands of optimization methods in the literature: each numerical method can solve a specific or more generic problem. Different algorithms are intended to solve different types of multiobjective optimization problems such as linear, nonlinear, continuous, discrete, mixed, and so on. Different strategies can be selected for different problems. Unfortunately, real world applications often include one or more difficulties which make many of these methods inapplicable. Many engineering problems involve highly non-linear objective functions or even may not have an analytic expression in terms of the variables. A general overview of basic and recent approaches to multiobjective optimization has been given in Chapters 1–7.

Therefore, a multipurpose software that can be used in several fields and contests should include the most widely used and state-of-the-art methods using both *MCDM based* and *metaheuristics* approaches to multiobjective optimization. Obviously, some specific problems can be solved with software that contain only few mathematical programming based methods. Unfortunately, decision makers or analysts do not necessarily know the mathematical formulation of the problem at hand and the problem can change time after time. These are the main reasons why a really multipurpose software represents a viable solution.

12.2.3 Visualization, Post-processing and Statistical Charts

Visualization is the key in understanding the results coming out from large simulations in computational science and engineering. After a multiobjective optimization, we typically wish to visualize the entire set of results, rather than simply analyzing each single result. Understanding the results of a multiobjective process can be quite hard, particularly in higher dimensional spaces. Even though there are plenty of generic visualization tools (such as 2D and 3D scatter plots as explained in Chapter 8), an ad hoc visualization tool for Pareto optimal solutions is needed. Visualizing the objective space and the Pareto points is quite easy with 2 or 3 objectives. For a higher number of objectives, some more complex techniques should be implemented. For example, a common way of visualizing multivariate problems is using a parallel coordinates chart (Inselberg and Dimsdale, 1990). Some more complex techniques can be really useful with high dimensional spaces. Two important multi-dimensional visualizing tools are *self organizing maps* (SOMs) (Kohonen, 1982) that can really speed up the optimization phase as reported in (Obayashi and Sasaki, 2004) and *heatmaps* as described by Pryke *et al.* (2007).

These visualizing tools should be considered even as tools for *data management* and preliminary exploration. In multiobjective optimization, an initial explorative phase, called as a learning phase in Chapter 2, is important in order to determine the behavior and the main characteristics of the problem at hand. The principal aim of a preliminary exploration is to get the most relevant qualitative information from a problem making the smallest possible number of evaluations. This can be done by using a smart positioning of points in the space. This methodology provides a strong tool to design and analyze functions; it eliminates redundant observations and reduces the time and resources to make evaluations and experiments (Fig. 12.2).

Moreover, traditionally, visualization and statistical charts have been used as post-processing operations to visualize results. Anyhow, visualization can also be used to show the quality of the solutions. This kind of visualization can

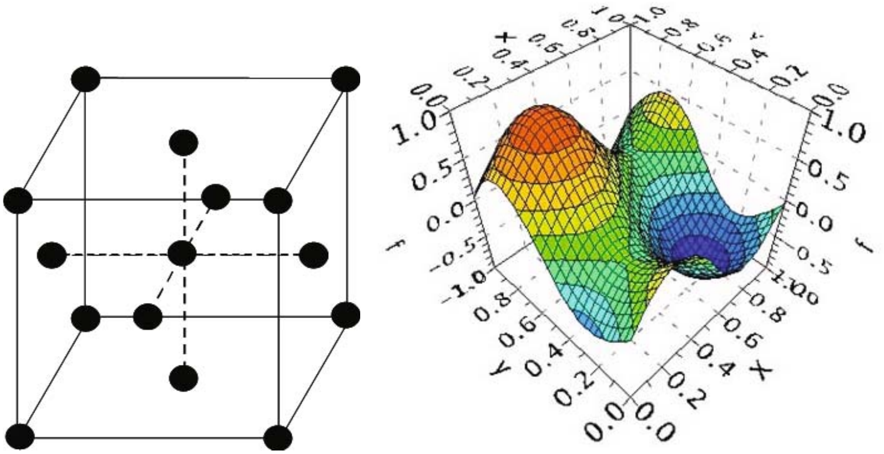


Fig. 12.2. Data management and preliminary exploration methods. A smart positioning of points in a 3-dimensional space (left) and a reliable meta-model (right)

give an important feedback during runtime and a good chart can support in deciding whether the optimization is going in the right direction or not. Based on visual feedback, the decision maker can stop and re-run the optimization using different parameters.

More detailed discussion of visualizing multiobjective optimization results is given in Chapters 8 and 9.

12.2.4 Decision Support Tool

In the absence of preference information, all Pareto optimal solutions can be regarded as equally desirable in the mathematical sense. Ranking a long list of Pareto optimal or nondominated alternatives is a difficult task especially when several solutions are available or when several conflicting goals are involved.

In several cases, more than one decision maker can be involved in selecting the best solution. In these cases, each person may even reflect different competencies and roles. Therefore, making coherent choices, with rational and transitive preferences, can be a very difficult task.

A decision support tool can assist the decision maker(s) in finding the best solution from among a set of reasonable alternatives. Moreover, a decision support tool can even allow the correct grouping of objectives into a single utility function by identifying possible relations between the objectives. A decision support tool can even guide the DM(s) in specifying preferences which leads to constructing a scalarized function that results to be coherent with the given preferences (see Chapters 1 and 2).

12.2.5 Meta-modeling and Validation of Models

In real life applications, it is not always possible to reduce the complexity of the problem and obtain a function that can be evaluated quickly. As reported in Chapter 11, in many practical engineering design and other scientific optimization problems, every single function evaluation can take hours or even days. In these cases, the time to run a single step of an algorithm makes running more than a few evaluations prohibitive and some other smart approaches are needed. In these situations, decision makers can turn to a preliminary exploration technique to perform a reduced number of calculations. After that, it is possible to use these well-distributed results to create a surface which interpolates these points. This surface represents a meta-model of the original problem and can be used to perform the optimization without costly computations. The use of mathematical and statistical tools to approximate, analyze and simulate complex real world systems is widely applied in many scientific domains. These kinds of interpolation and regression methodologies are now becoming common even for solving complex optimization problems where they are also known as *response surface methods* (RSMs). For example, RSMs are becoming very popular offering a surrogate model with a second generation of improvements in speed and accuracy in computer aided engineering. This approach allows direct optimization otherwise impossible.

Constructing a useful meta-model starting from a reduced number of real evaluations is not a trivial task. Mathematical and physical soundness, computational costs and prediction errors are not the only points to take into account when developing meta-models. Ergonomics of the software has to be considered in a wide sense. The users would like to grasp the general trends in the phenomena, especially when the behavior is nonlinear. Moreover, de-

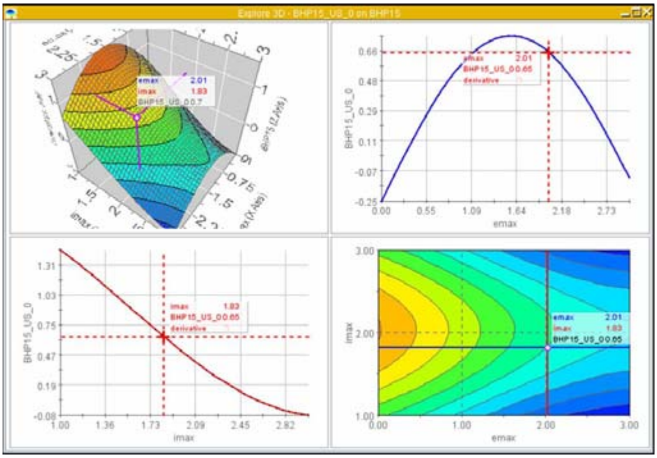
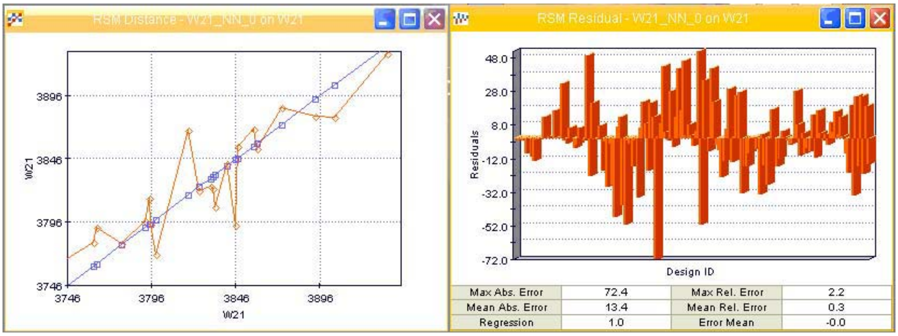


Fig. 12.3. A tool for meta-models: 3D-exploration

cision makers and engineers would like to re-use the experience accumulated, in order to spread the possible advantages to different projects. When using meta-models, the users should always keep in mind that this instrument allows a faster analysis than the complex models, but interpolation and extrapolation introduce a new element of error that must be managed carefully.

For these reasons, in the last years, different approximation strategies have been developed to provide inexpensive meta-models of the simulation models to substitute computationally expensive modules. As reported in Chapter 11, there is not a unique meta-model that is valid for any kind of situations. For this reason a good multiobjective optimization software should contain several different interpolation techniques such as, for example, neural networks, radial basis functions, kriging and Gaussian processes (see Chapter 11).

Once the meta-model has been constructed, it is really important to certify its fidelity. This is the reason why a tool for exploring (Fig. 12.3) and measuring the quality (Fig. 12.4) of meta-models in terms of statistical reliability would be appreciated together with all approximation strategies.



Max Abs. Error	72.4	Max Rel. Error	2.2
Mean Abs. Error	13.4	Mean Rel. Error	0.3
Regression	1.0	Error Mean	-0.0

Fig. 12.4. Tools for measuring the quality of meta-models. Distance chart that points out the differences between real values and values calculated using the meta-model, (left) and residual chart (right). The residuals are the amounts which the meta-model has not been able to explain (approximation errors). These charts help to determine whether a meta-model is an acceptable representation of the original problem.

12.2.6 Robustness and Reliability

When dealing with uncertainty, conventional optimization techniques tend to “over-optimize”, producing solutions that may perform well at the optimal point but have poor characteristics against the dispersion of design variables or environmental variables. As reported in Chapter 16, it is quite possible that the optimal solution will not be the most stable solution. For example, the function in Fig. 12.5 has a global optimum at point A, and a local optimum at point B. However, any small variations in the input parameters will cause the performance to drop off markedly around A. The performance of B may not be as good in absolute terms, but it is much more robust, since small changes in the input values do not cause drastic performance degradation.

For this reason, a tool that allows the user to perform a robust design analysis to check on the system's sensitivity to manufacturing tolerances or small changes in operating conditions can be really useful.

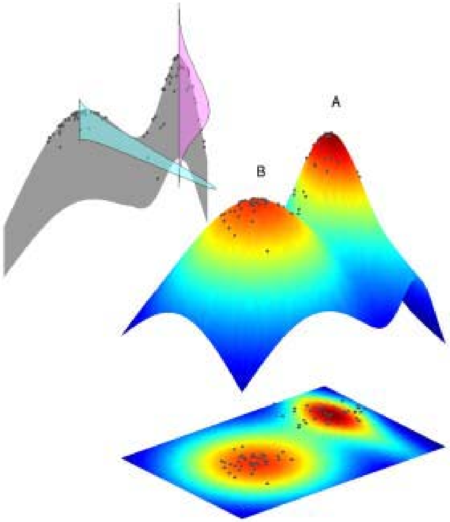


Fig. 12.5. Robustness of solutions

12.2.7 Parallelization

In those cases where each single function evaluation can be really time-consuming, *parallel computing* can be an important resource. In few words, *parallel computing* refers to evaluating simultaneously the same function on several processors in order to obtain results faster. The simple idea of parallelization is based on the fact that the optimization process usually can be divided into smaller steps. These smaller steps can be carried out simultaneously on multiple computers with some special coordination. The coordination can be done by a central manager that manages all the computers of the pool, collecting the requests and moving the computations accordingly to the current load of each computer. In this way, the whole optimization, or a part of it, can even be submitted to a queuing system and executed taking advantages of several different remote computers. This concept has been well described in Chapter 13.

This approach can really speed up the optimization because a parallel optimization algorithm can be much faster than a corresponding sequential algorithm. Parallel optimization methods can be developed by redesigning serial algorithms to make effective use of parallel hardware. Unfortunately, not all algorithms can be parallelized: for example, evolutionary algorithms can be parallelized more easily than many MCDM approaches.

12.2.8 Plug-in

A very good quality for a software is to be a completely open platform where anyone can contribute. The complexity of multiobjective optimization is becoming too big to design monolithic platforms. That is the reason why an open platform where scientists and software engineers can introduce their own methodologies and algorithms may represent a good solution.

Open platforms usually provide application programming interfaces (APIs) allowing third parties to create plug-ins that interact with the main application. In this kind of an open platform, the users can contribute with their own optimization techniques without any changes to the main platform. Using the APIs the users can introduce the optimization technique that is most appropriate for solving the problem at hand. An example of an open framework dedicated to the design of metaheuristics is Paradiseo-MOEO (Liefooghe *et al.*, 2008).

12.3 List and Description of Software

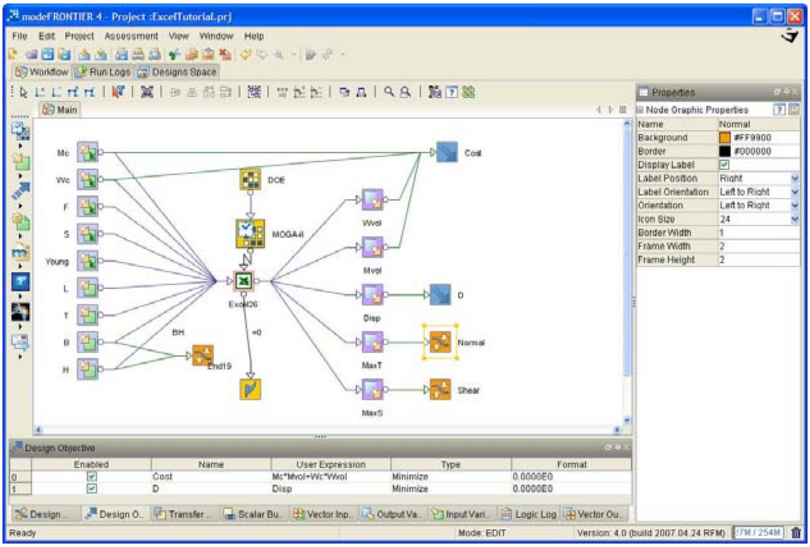
Several software cover one or at most two of the previously discussed properties. There are several multiobjective optimization tools available; and each tool can solve a specific or more generic problem. Some tools are more appropriate for constrained optimization, others may be suitable for unconstrained continuous problems, or *tailored* for solving some specific problems. Unfortunately, real world applications often include one or more difficulties which make these tools inapplicable. Most of the time, objective functions are highly nonlinear or even may not be given in a closed form in terms of the design variables.

In this chapter, we describe only *general purpose* software tools that have been built from the ground up to solve multiobjective optimization problems. Therefore, we describe software that can efficiently handle several goals and constraints at the same time, allowing to choose the best solution from a set of solutions that represent the best trade-offs.

In this section we identify and select a collection of free and commercial general purpose software. An expanded list of other software and interesting libraries is also provided in Section 12.5.

12.3.1 modeFRONTIER

modeFRONTIER is a multiobjective optimization and design environment, written to allow easy coupling to almost any computer aided engineering (CAE) tool. modeFRONTIER provides an environment which allows product engineers and designers to integrate their various CAE tools, such as CAD, finite element structural analysis and computational fluid dynamics (CFD) software. There are also direct interfaces for Excel, Matlab and Simulink. Using a variety of state-of-the-art multiobjective optimization techniques, ranging from gradient-based methods to genetic algorithms, the process or design of interest can be optimized by specifying objectives and defining variables which affect factors such as geometric shape and operating conditions. modeFRONTIER (Fig. 12.6) in effect becomes a wrapper around the CAE tool, performing the optimization.



Node Graphic Properties

Name	Normal
Background	#FF9900
Border	#000000
Display Label	☑
Label Position	Right
Label Orientation	Left to Right
Orientation	Left to Right
Icon Size	24
Border Width	1
Frame Width	2
Frame Height	2

	Enabled	Name	User Expression	Type	Format
0	☑	Cost	Mc*Mvol+Wc*Wvol	Minimize	0.0000E0
1	☑	D	Disp	Minimize	0.0000E0

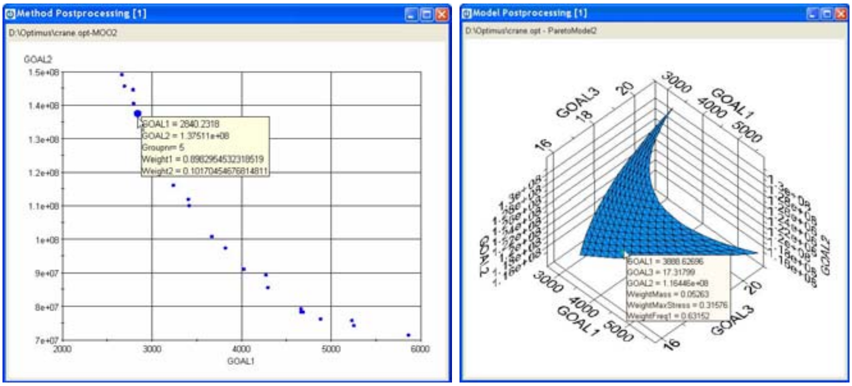
Fig. 12.6. A snapshot of modeFRONTER graphical users interface. In this panel the user can define the optimization problem.

modeFRONTIER includes a wide range of possible algorithms that can be selected for solving different problems. At present, the multiobjective methods available in modeFRONTIER are: Multiobjective Genetic Algorithm (MOGA), Adaptive Range MOGA, Multiobjective Simulated annealing (MOSA), Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb *et al.*, 2002), Multiobjective Game Theory, Evolutionary Strategies Methodologies and Normal Boundary Intersection (NBI) (Das and Dennis, 1998). Moreover, different algorithms can even be combined by the decision makers in order to obtain some hybrid approaches according to their preferences. A 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. modeFRONTIER is a commercial software developed by ESTECO; its website contains several examples of how to use the software for solving multiobjective optimization problems and decision making processes in engineering.

12.3.2 OPTIMUS

OPTIMUS is a world-leading process integration software, that bundles a collection of design exploration and numerical optimization methods. It allows users to build simulation workflows to automate their numerical simulation processes. These simulation workflows integrate one or more simulation codes and are executed by OPTIMUS - if possible - without user intervention. Once the simulation process is captured in an OPTIMUS workflow, users are able to explore their design space by modifying selected input parameters and hunt for new designs that are more reliable with better functional performance. All calculations are based on the integrated simulation tools that are part of the OPTIMUS workflows.

Methods available in OPTIMUS include:



Method Postprocessing [1]

D:\Optimus\crane opt-MOO2

GOAL1 = 2840.2318
GOAL2 = 1.37511e+08
Groupnr= 5
Weight1 = 0.8982954532318519
Weight2 = 0.10170454676814811

Model Postprocessing [1]

D:\Optimus\crane opt - ParetoModel2

GOAL1 = 3888.62696
GOAL3 = 17.31799
GOAL2 = 1.16446e+08
WeightMass = 0.05263
WeightMaxStress = 0.31576
WeightFreq1 = 0.63152

Fig. 12.7. Optimization Post-Processing with OPTIMUS. A scatter plot showing points on the Pareto optimal set for two objectives. When clicking on a point, the variable values are shown (left). A 3-dimensional plot of the Pareto optimal set (right).

- design space exploration, such as Design of Experiments (DOE) and Response Surface Modeling (RSM),
- numerical optimization, based on gradient-based local algorithms or genetic global algorithms, both for single or multiple objectives with continuous and/or discrete design variables and
- robustness and reliability engineering, including methods to assess and optimize the variability of design outputs based on variable design inputs.

The Multiobjective optimization methods include: non-dominated sorting evolutionary algorithm (NSEA and NSEA+, based on NSGA-II), normal-boundary intersection method as well as the weighting method, the weighted Chebyshev problem, the trade-off method, lexicographic ordering and the method of global criterion (see, for example, Chapters 1 and 2).

OPTIMUS is developed by Noesis Solutions, a subsidiary of LMS International, headquartered in Leuven, Belgium.

12.3.3 iSIGHT

Engineous' iSIGHT software integrates and manages the computer software required to execute simulation-based design processes, including commercial CAD/CAE software, internally developed programs, and Excel spreadsheets. iSIGHT drives toward optimal and reliable product designs using the library of advanced engineering tools. iSIGHT components include: optimization, design of experiments, Monte Carlo analysis, approximations. iSIGHT is continuously updated and contains state-of-the-art multiobjective genetic algorithm routines such as for example MOGA-NCGA and NSGA-II.

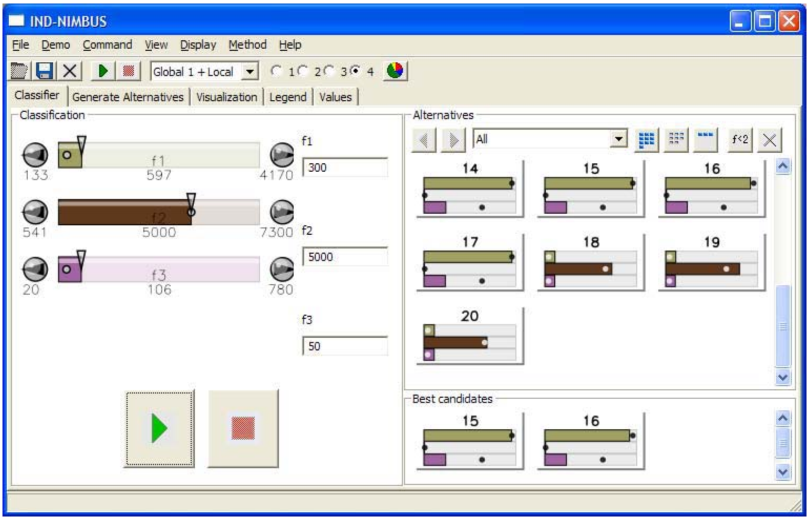
12.3.4 NIMBUS

NIMBUS (Miettinen and Mäkelä, 2006) is an interactive classification-based method for multiobjective optimization, see Chapter 2 (Miettinen, 1999). It is suitable for both differentiable and nondifferentiable multiobjective and single objective optimization problems subject to nonlinear and linear constraints with bounds for the variables. The classification information obtained from a decision maker is used to generate one to four Pareto optimal solutions that best reflect the preferences expressed in the classification. In practice, this means that one to four subproblems are created and solved with a solver appropriate for the characteristics of the problem in question. The subproblems include also reference point based subproblems from the reference point method, the STOM and the GUESS methods (see Chapter 2). WWW-NIMBUS is the implementation of the NIMBUS method operating via the Internet at `http://nimbus.it.jyu.fi/`. WWW-NIMBUS can be used free of charge for academic purposes. The implementation operating under Linux and MS-Windows operating systems is IND-NIMBUS (Miettinen, 2006). It is for sale and some information about it is available at http://ind-nimbus.it.jyu.fi/. An example of the user interface of IND-NIMBUS (classification window) is given in Figure 12.8.

In both the implementations, there are several underlying solvers available including a local proximal bundle method and a global genetic algorithm with different constraint-handling techniques. It is also possible to use a hybrid solver where the local solver is used after the global one. Both the implementations support the DM in comparing Pareto optimal solutions (s)he likes with graphical visualizations. Besides classification, the DM can also direct the search by asking for intermediate solutions between any two generated solutions.

12.3.5 PROMOIN

The interactive system PROMOIN (Caballero *et al.*, 2002) has been designed as a decision aid tool for multiobjective problems, based on the use of interactive techniques. The current version of the software deals with linear problems, although a nonlinear version is currently under construction. The main idea underlying PROMOIN is the following. There are plenty of interactive techniques available in the literature. They differ in several aspects: the kind of problem handled, the type of final solution, the inner solution process and the information asked from the decision maker. The last issue is a key factor for the success of an interactive method. If the decision maker does not feel comfortable with the information (s)he has to provide, the method will hardly succeed in finding her/his most preferred solution. Therefore, the interactive technique should be chosen according to the decision maker's wishes



Global 1 + Local

f1 300

f2 5000

f3 50

Fig. 12.8. IND-NIMBUS: graphical users interface

regarding how (s)he wants to give the information. Furthermore, the kind of information that the decision maker wishes to give may vary during the solution process, due to the fact that (s)he progressively learns about the problem, and gets a more accurate picture of the problem. The main interactive procedures have been incorporated into the system, including tradeoff based methods such as the Geoffrion-Dyer-Feinberg (GDF) method, the sequential proxy optimization (SPOT) method, the interactive surrogate worth trade-off method (ISWT) and other methods such as the Tchebycheff method and the Zionts-Wallenius method. Moreover, it contains reference point based methods such as STEM, STOM and the reference point method. (For further details of the methods, see, e.g., Chapter 2.) The method can be chosen according to the decision maker's wishes. On the other hand, the system offers the possibility to change between methods any time during the solution process. The program has been implemented under Windows environment, with the aim of providing the user with a friendly interface.

12.3.6 MKO-2

The interactive software system MKO-2 is a generalized multiobjective decision support system (Staykov, 2006; Vassilev *et al.*, 2006). It has been designed to support solving linear and linear integer multiobjective optimization problems. The system implements an innovative generalized classification-based interactive algorithm for multiobjective optimization with variable scalarizations and parameterizations, applicable for different types of problems (i.e., linear, nonlinear and mixed variables). The MKO-2 system will be extended to handle nonlinear multiobjective optimization problems as well. The incorporated generalized interactive algorithm is applicable for different ways of defining the preferences by the decision maker, such as weighting factors (priorities), aspiration levels, aspiration intervals, aspiration directions of change in the values of some or of all the objective functions, etc. Using the MKO-2 system, the decision makers can apply twelve interactive MCDM methods existing in the literature (the Chebyshev method, the STEM method, the STOM method, the reference point method, the GUESS Method (Buchanan, 1997), the modified reference point method, the visual interactive method, the reference direction method, the NIMBUS method, the DALDI method, the weighting method, and the ϵ-constraint method; see Chapters 1 and 2) and different strategies in the search for new Pareto optimal solutions, not only with the help of one particular method, but combining different interactive MCDM methods. In this way, the MKO-2 software system can be used not only for solving multiobjective optimization problems, but also for comparing and analyzing different solutions of a given problem, using different types of preference information, set by the decision maker, and different interactive methods. The MKO-2 software system operates under MS Windows operating system. The graphical user interface of the system enables decision makers with different degrees of qualification, referring to the methods and software tools, to operate easily with the system. MKO-2 decision support system can be used both for education and for solving real-life problems. It can be used free for academic purposes under a certain bilateral agreement.

12.3.7 Pareto Front Viewer

The software "Pareto Front Viewer" (PFV) provides interactive visualization of the Pareto frontier for multiobjective problems in the case of two to eight objective functions. It is assumed that an approximation of the Pareto frontier has already been constructed in the form of a finite list of objective vectors. The method for constructing the objective vectors plays no role. Thus, the PFV software can be combined with any Pareto frontier approximation technique. The method is based on the visualization of bi-objective slices of the Edgeworth-Pareto Hull (EPH) of the objective vectors, that is, the union of the domination cones with vertices located in the objective vectors of the approximation. The objective tradeoffs for any three objectives that are specified by a decision maker are visualized in the form of decision maps, which are collections of the overlaid bi-objective slices of the EPH. The influence of the other objectives can be studied by moving the sliders of the related scroll-bars. By this, the user is informed on the objective tradeoffs and is supported in the process of selecting the preferred Pareto optimal decision vector, which is based on the direct identification of the feasible goal at the computer screen. The software PFV was coded for the platforms MS Windows 98/NT/2000/XP. A demonstration version for up to five criteria and 500 criterion points can be downloaded for free from `http://www.ccas.ru/mmes/mmeda/soft/third.htm`.

12.3.8 Reasonable Goals Method for Databases

Reasonable Goals Method for Databases (RGDB) is the Web application server that supports selecting a small number of alternatives from large tables through Internet. The application server applies the Reasonable Goals Method, that is, visualization of the Pareto frontier of the envelope (convex hull) of the objective vectors related to the alternatives in the case of two to eight objectives. First, the user provides the table with alternatives to the server. Then, the server approximates the Edgeworth-Pareto Hull (EPH) of the convex hull of the objective vectors. Then, the user's computer receives the applet that supports interactive visualization of the Pareto frontier based on the visualization of bi-objective slices of the EPH. The user explores objective tradeoffs for any three selected objectives. The influence of the other objectives can be studied by moving the sliders of the related scroll-bars. Then, the reasonable goal is transmitted to the server, which selects a small number of the Pareto optimal decisions and transmits them back to the user. The Web application server coded in C++ can be used with the help of standard browsers. A demonstration version (5 objectives and up to 500 alternatives), can be found at `http://www.ccas.ru/mmes/mmeda/rgdb/index.htm`.

12.3.9 ParadisEO and GUIMOO

ParadisEO is a white-box object oriented generic framework dedicated to the flexible design of evolutionary multiobjective algorithms. This paradigm-free software aims to provide a set of classes allowing to ease and speed up the development of computationally efficient programs. It is based on a clear conceptual distinction between the solution methods and the multiobjective problems they are intended to solve. This separation confers a maximum design and code reuse. ParadisEO provides a broad range of archive-related features (such as elitism or performance metrics) and the most common Pareto-based fitness assignment strategies such as MOGA, NSGA, SPEA and Indicator-Based Evolutionary Algorithm (IBEA) (Zitzler and Künzli, 2004). Furthermore, parallel and distributed models as well as hybridization mechanisms can be applied to an algorithm designed within ParadisEO. This tool is developed by INRIA that, in addition, provides GUIMOO, a platform-independent free software dedicated to analysis of results of multiobjective problems. GUIMOO allows visualization of approximative Pareto frontiers and contains metrics for quantitative and qualitative performance evaluations.

12.4 Summary Table on Optimization Software

Table 12.1 summarizes the main characteristics of all the multiobjective optimization software tools described in the previous section. This table lists only tools that have been developed exclusively for multiobjective optimization. Many other software systems can be used for optimization or visualization of data but we limit our study to the tools that look for the Pareto frontier and have visualization capabilities dedicated to this type of results.

The table has 11 columns that can be read as follows:

1. **Software and Developers**: name of the software and information about companies or institutions taking care of the development. Whenever possible, web-pages or email contacts are reported.
2. **Platforms**: platforms where the software can run
3. **An easy-to-use GUI**
4. **EMO**: this column contains a brief description of the evolutionary multiobjective methods available in the software.
5. **MCDM**: This column contains a brief description of the MCDM methods available in the software.
6. **Rob.**: This column contains the symbol X if and only if the software contains at least a method to establish robustness of solutions.
7. **Meta**: This column reports whether or not the software contains one or more methods for meta-modeling.
8. **Vis**: visualization tool and statistical analysis. This column indicates whether or not the software contains one or more methods for visualizing the Pareto frontier and/or other results coming out from the optimization phase.

9. **Plug**: this column contains the symbol X if the software can be considered as an open platform where the users can add their own optimization methods as external plug-ins.
10. //: this symbol stands for **Parallelization**. Hence this column contains the symbol X if the software supports parallel computation and if the optimization algorithms can deal with queuing systems.
11. **License**: License type, commercial, free or academic.

Table 12.1. Summary of the main characteristics of the multiobjective optimization software described

Software and Developers	**Platforms**	**GUI**	**EMO**	**MCDM**	**Rob.**	**Meta**	**Vis.**	**Plug**	**License**	‖
modeFRONTIER ESTECO www.esteco.com	All	x	MOGA-II, NSGA-II, ARMOGA, MOSA, Game Theory, MO evolutionary strategies	NBI, weighting method, reference point method	x	x	x	x	Commercial	x
OPTIMUS NOESIS Solutions www.noesissolutions.com	Windows and Linux	x	Non-Dominated Sorting Evolutionary Algorithms (NSEA and NSEA+)	NBI and 7 other methods	x	x	x	x	Commercial	x
iSIGHT Engineous Software www.engineous.com	All	x	NCGA, NSGA-II	LSGRG, MMFD, MOST, Stress Ratio, MIGA, Pointer	x	x	x	x	Commercial	x
WWW-NIMBUS, University of Jyväskylä, nimbus.it.jyu.fi	Web	x		NIMBUS, reference point method, GUESS, STOM			x		Free for Academic	
IND-NIMBUS, University of Jyväskylä, ind-nimbus.it.jyu.fi	Windows and Linux	x		NIMBUS, reference point method, GUESS, STOM			x		Commercial	
PROMOIN University of Malaga	Windows	x		SPOT, ISWT, STEM, STOM					Free	
MKO-2 Institute of Information Technologies - Bulgarian Academy of Sciences, Department of Decision Support Systems www.iit.bas.bg	Windows	x		Chebyshev, STEM and STOM methods, GUESS, reference point method, NIMBUS, DALDI, weighting method			x		Free for Academic	
Pareto Front Viewer www.ccas.ru/mmes/mmeda/soft	Windows	x					x	x	Commercial, Free up to 500 points	
Reasonable Goals Methods for Databases www.ccas.ru/mmes/mmeda/rgdb/	Web	x					x	x	Free	
ParadisEO Inria paradiseo.gforge.inria.fr	All	x	NSGA-II, IBEA					x	Commercial	x

12.5 List of Available Libraries

Much of the evolutionary multiobjective optimization studies use computer codes which are freely downloadable. Some of them are the NSGA-II code in C (http://www.iitk.ac.in/kangal/soft.htm), SPEA2 and other EMO codes in C++. An important platform containing a set of ready-to-go multiobjective optimization methods is *PISA* (Bleuler *et al.*, 2003). PISA consists of two parts: a set of optimization problems (variators) and a set of optimization algorithms (selectors). The selectors are state-of-the-art evolutionary multiobjective optimization methods (see Chapter 3). The user can write and submit a new module in the platform. All modules available can be used for academic purposes without a fee. Each module specifies its own licensing policy. PISA itself is a copyright of the Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory.

There are several other platforms that help the development process of evolutionary multiobjective optimizers, for example, *Open BEAGLE* which is an Object-Oriented software environment enabling the implementation of almost any kind of evolutionary algorithm, such as genetic algorithms and genetic programming, MOMHLib++, MOEA (Tan *et al.*, 2000).

Another important package is *DAKOTA*, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis developed by the Sandia National Laboratories.

For MCDM based multiobjective methods there are several items available in the Internet such as, for example, PROTASS developed by Rafal Cytrycki for linear multiobjective problems and available for download at *http://www.ekspert.szczecin.pl/protass/en.*

Even though designed for discrete problems, let us still mention one of the most famous decision support systems, the Analytic Hierarchy Process (AHP). Designed to reflect the way people actually think, AHP was developed in the 1970's (Saaty, 1996). AHP is now included into a commercial software called *Expert Choice*. This software is intuitive, graphically based and structured in a user-friendly fashion. With this tool, decision makers are able to drill down to their level of expertise, and apply judgments to the objectives to achieving their goals.

Finally, we can mention the Decisionarium project (http://www.decisionarium.net/) which focuses on the development of web based tools for interactive multicriteria decision support for individual decision making, for group collaboration and negotiation as well as for interaction and surveys over the web (mostly for discrete problems).

There are probably several other packages that should be listed in this paragraph. Unfortunately, most of the times, software products are implemented for academic testing purposes and are usually neither maintained nor advertised. Moreover, there exist several tools for solving single objective optimization problems that may contain some possibilities for multiobjective optimization. Those are deliberately excluded from this final list because this chapter wants to concentrate only on nonlinear multiobjective optimization software developed in the last decade.

12.6 Conclusions

The information collected and presented in this chapter is just a snapshot of the multiobjective optimization tools available. Setting up, installing and testing all these software packages on a number of different platforms has been a quite demanding job. It is obviously impossible to say which one is the best amongst all the listed software. A high number issues should be taken into account for evaluating a software such as ease of use, completeness, configurability, robustness, efficiency, user support and so on.

Software for multiobjective optimization and more complex integrated environments such as “Process Integration and Design Optimization” (PIDO) or “Problem Solving Environment” (PSE) have become popular in the last years and several new packages are probably coming on the market. The importance of multiobjective optimization for the commercial world can be readily seen by the fact that most of the industrial companies now support one or more of the available packages. There is clear evidence that both commercial and research/academic communities are becoming increasingly interested in multiobjective optimization software.

The data summarized in Table 12.1 let us conclude that commercial software are usually more complete and more close to an advisable multiobjective optimization software than a free or open sources tool. Anyhow, there are even some good libraries that are well-qualified starting points for people approaching multiobjective optimization.

References

Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A Platform and Programming Language Independent Interface for Search Algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)

Buchanan, J.T.: A Naiive Approach for Solving MCDM Problems: The GUESS Method. The Journal of the Operational Research Society 48(2), 202–206 (1997)

Caballero, R., Luque, M., Molina, J., Ruiz, F.: Promoin: An interactive system for multiobjective programming. Information Technologies and Decision Making 1, 635–656 (2002)

Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. SIAM Journal on Optimization 8(3), 631–657 (1998)

Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 181–197 (2002)

Gallopoulos, E., Houstis, E., Rice, J.R.: Future Research Directions in Problem Solving Environments for Computational Science (1991)

Houstis, E., Gallopoulos, E., Bramley, R., Rice, J.: Problem-Solving Environments for Computational Science. IEEE Computational Science and Engineering 4(3), 18–21 (1997)

Inselberg, A., Dimsdale, B.: Parallel Coordinates: a Tool for Visualizing Multi-Dimensional Geometry. In: VIS '90: Proceedings of the 1st conference on Visualization '90, San Francisco, California, pp. 361–378. IEEE Computer Society Press, Los Alamitos (1990)

Kaliszewski, I.: Out of the mist – Towards decision-maker-friendly multiple criteria decision making support. European Journal of Operational Research 158, 293–307 (2004)

Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 59–69 (1982)

Liefooghe, A., Basseur, M., Jourdan, L., Talbi, E.-G.: ParadisEO-MOEO: A Framework for Evolutionary Multi-objective Optimization (2008)

Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)

Miettinen, K.: IND-NIMBUS for demanding interactive multiobjective optimization. In: Trzaskalik, T. (ed.) Multiple Criteria Decision Making '05, pp. 137–150. Karol Adamiecki University of Economics, Katowice (2006)

Miettinen, K., Mäkelä, M.M.: Synchronous approach in interactive multiobjective optimization. European Journal of Operational Research 170(3), 909–922 (2006)

Obayashi, S., Sasaki, D.: Multi-objective optimization for aerodynamic designs by using armogas. In: HPCASIA '04: Proceedings of the High Performance Computing and Grid in Asia Pacific Region, Seventh International Conference on (HPCAsia'04), Washington, DC, USA, pp. 396–403. IEEE Computer Society Press, Los Alamitos (2004)

Pryke, A., Mostaghim, S., Nazemi, A.: Heatmap Visualization of Population Based Multi Objective Algorithms. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 361–375. Springer, Heidelberg (2007)

Saaty, T.L.: Multicriteria Decision Making: The Analytic Hierarchy Process; Planning, Priority Setting, Resource Allocation, 2nd edn. Analytic Hierarchy Process Series. RWS Publications, Pittsburgh (1996)

Staykov, B.: Multiobjective optimization software system. In: Problems of Engineering Cybernetics and Robotics, vol. 57, pp. 21–30 (2006)

Tan, K.C., Lee, T.H., Khoo, D., Khor, E.F.: MOEA Toolbox for Computer-Aided Multi-Objective Optimization. In: 2000 Congress on Evolutionary Computation, July 2000, vol. 1, pp. 38–45. IEEE Computer Society Press, Piscataway (2000)

Vassilev, V., Vassileva, M., Staykov, B., Miettinen, K.: Generalized multicriteria decision support systems. In: Proceedings of the International Workshop on Semantic Web and Knowledge Technologies Applications, 12th International Conference AIMSA, pp. 16–30 (2006)

Weistroffer, H.R., Smith, C.H., Narula, S.C.: Multiple criteria decision support software. In: Figueira, J., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 989–1018. Springer, Heidelberg (2005)

Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004)