Kaisa Miettinen

IND-NIMBUS FOR DEMANDING INTERACTIVE MULTIOBJECTIVE OPTIMIZATION*

INTRODUCTION

In multiobjective optimization, we handle optimization problems with more than one objective function. Because the objectives are typically conflicting, they cannot reach their individual optima simultaneously but the aim is to find the best compromise. As compromises, we define mathematically equivalent nondominated or Pareto optimal solutions. To be able to select the best possible compromise, we usually need the input of a human expert, a decision maker who can express preference information related to the problem.

A widely used approach for solving multiobjective optimization problems is to use interactive methods, where the decision maker can iteratively direct the solution process and, at the same time, learn about the interdependencies among the objectives in the problem to be solved (see e.g. [2; 9; 10; 14; 26; 29] and references therein). Interactive methods differ from each other, for example, by what kind of information is to be given to the decision maker about the problem, what kind of preference information is asked from the decision maker and how the preference information is used for directing the solution process. In some interactive methods, the decision maker can even change her/his mind while learning. Besides giving the decision maker a possibility to learn about the problem, interactive methods are usually computationally efficient because only such Pareto optimal solutions are generated that the decision maker is interested in. For these reasons, interactive methods can be expected to give very satisfactory solutions. This naturally necessitates that the decision maker has enough time and interest to take part in the iterative solution process.

Unfortunately, software for solving multiobjective optimization problems involving continuous variables is not easy to find. The task gets even more difficult if the problem is nonlinear. WWW-NIMBUS [17], an interactive software system operating on the Internet, was developed in 1995 to answer this need. WWW-NIMBUS (available at http://nimbus.it.jyu.fi/) has changed quite

*The implementation of IND-NIMBUS was realized in a project supported by the Finnish Funding Agency for Technology and Innovation. The author wishes to thank Dr. Marko M. Mäkelä for his share in developing NIMBUS as well as the other members of the team that worked in the above-mentioned project.

a lot during the years but it can still be used free of charge for teaching and academic research proposes. WWW-NIMBUS can be used for solving nonlinear and even nondifferentiable and nonconvex multiobjective optimization problems. Because the Internet is easily accessible, the system is automatically available to large numbers of people. In 1995, the first version of WWW-NIMBUS was the first interactive multiobjective optimization software operating on the Internet. Even now, it continues to be a unique software system.

WWW-NIMBUS is based on the principles of centralized computing and distributed interface. This means that all the calculations take place in a server computer (at the University of Jyväskylä) and the user interface is the browser of each individual user. In this way, the system sets no requirements on the user's computer and the operating system used and/or compilers available play no role. There is nothing to be installed and the latest version of the system is always available. Furthermore, the World-Wide Web (WWW) provides a convenient and graphical user interface with visualization possibilities.

Operating via the Internet is convenient with academic problems but when the problem involves computationally expensive function evaluations and/or the functions values come from a simulation or modelling tool, another approach is needed. For this purpose, IND-NIMBUS (INDustrial NIMBUS), a software operating in MS-Windows and Linux operating systems has been developed.

The NIMBUS method is the core of both WWW-NIMBUS and IND-NIMBUS. NIMBUS (Nondifferentiable Interactive Multiobjective BUndle-based optimization System) is an interactive method where preference information is acquired from the decision maker in the form of a classification of the objective functions. The method has been applied, for example, in structural design problems [21], in the optimal control problems of the continuous casting of steel [22] and in the optimal shape design of paper machine headboxes [7]. Results with both small-scale and large-scale problems give evidence of the reliability and efficiency of the method. Different versions of NIMBUS are described in [14; 15; 16]. Here we concentrate on the latest, so-called synchronous, version [19].

In NIMBUS, the decision maker can iteratively learn about the problem and can conveniently direct the solution process. NIMBUS has been designed to be easy to use and, unlike many interactive methods, it does not require consistent information from the decision maker. Furthermore, the information handled is straightforward. The objective function values have a direct meaning to the decision maker and no artificial concepts are needed.

In this paper, we briefly introduce the synchronous NIMBUS method in Section 1 and its implementation, IND-NIMBUS in Section 2. Finally, we conclude in Section 3.

1. NIMBUS METHOD

Multiobjective optimization problems to be considered are of the form

$$\begin{array}{rl} \text{minimize/maximize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S \end{array}$$

with k *objective functions* $f_i : \mathbf{R}^n \to \mathbf{R}$ to be optimized simultaneously. Each of them is either to be minimized or maximized. The *decision vector* $\mathbf{x}$ belongs to the (nonempty) *feasible set* S. The feasible region may consist of linear and nonlinear inequality and equality constraints as well as box constraints for the variables. The images of the feasible decision vectors are called feasible *objective vectors*. Without loss of generality we assume in this section that all the objective functions are to be minimized.

The idea of the interactive NIMBUS method is to move around the set of Pareto optimal solutions, where the value of an objective function can only be improved by allowing at least one of the others to impair (see e.g. [14)]. In order to help the decision maker in getting an impression of what is possible to achieve, we need information about the ranges of the feasible objective vectors in the Pareto optimal set. We refer to the vector containing the best values of each objective function as the *ideal objective vector* $\mathbf{z}^\star \in \mathbf{R}^k$. The vector with the worst values is a so-called *nadir objective vector* $\mathbf{z}^{\mathrm{nad}} \in \mathbf{R}^k$. Unfortunately, it is difficult to obtain but can be approximated (see e.g. [14] and references therein). In what follows, we assume that we have approximations of ideal and nadir values available for each objective function.

In the interactive solution process with NIMBUS, the decision maker can at each iteration indicate what kind of a solution would be more satisfactory than the current one with the help of a classification. Thus, the user can evaluate the problem to be solved and adapt one's preferences during the solution process in an iterative and flexible way. Let $\mathbf{x}^h$ stand for the Pareto optimal decision vector at the iteration h. We show the objective function values calculated at this point to the decision maker. Then the decision maker is asked to classify the objective functions into up to five classes for objective functions f_i whose values

– should be decreased ($i \in I^<$),

– should be decreased till some aspiration level $\bar{z}_i^h < f_i(\mathbf{x}^h)$ ($i \in I^\leq$),

– are satisfactory at the moment ($i \in I^=$),

– are allowed to increase till some upper bound $\varepsilon_i^h > f_i(\mathbf{x}^h)$ ($i \in I^>$),

– are allowed to change freely ($i \in I^\diamond$).

The difference between the first two classes is that the objective functions in the first class are to be minimized as far as possible but the functions in the second class only till the aspiration level specified. The decision maker is asked to specify the aspiration levels and upper bounds, if needed. Since improvement in the Pareto optimal set in any objective function value is possible only by allowing impairment in some other objective function, the classification is feasible only if $I^< \cup I^\leq \neq \emptyset$ and $I^> \cup I^\diamond \neq \emptyset$.

After the classification, a *subproblem* [19] is formed based on the information specified as:

$$\begin{array}{ll} \text{minimize} & \max\limits_{\substack{i\in I^< \\ j\in I^\leq}} \left[\dfrac{f_i(\mathbf{x}) - z_i^\star}{z_i^{\text{nad}} - z_i^{\star\star}} \dfrac{f_j(\mathbf{x}) - \hat{z}_j}{z_j^{\text{nad}} - z_j^{\star\star}}\right] + \rho \displaystyle\sum_{i=1}^{k} \dfrac{f_i(\mathbf{x})}{z_i^{\text{nad}} - z_i^{\star\star}} \\ \text{subject to} & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^c) \quad \text{for all} \quad i \in I^< \cup I^\leq \cup I^=, \\ & f_i(\mathbf{x}) \leq \varepsilon_i \quad \text{for all} \quad i \in I^\geq \\ & \mathbf{x} \in S \end{array} \tag{1}$$

where a so-called augmentation coefficient $\rho > 0$ is a relatively small scalar. The weighting coefficients $1/(z_j^{\text{nad}} - z_j^{\star\star})$ have proven to facilitate capturing the preferences of the decision maker well. They also increase computational efficiency [20].

As shown in [18], different subproblems may lead to different solutions even though they are based on the same preference information. Usually method developers select one subproblem, which means that they select the solution to be generated. Yet, there is no general way how to identify the best solution without involving the decision maker.

In the synchronous version of NIMBUS [19], there are three subproblems available in addition to (1). This means that if the decision maker wants so, (s)he can see up to four different solutions after one classification. In other words, by classifying the objective functions once, the decision maker can get a better picture of different Pareto optimal solutions satisfying the preference information specified. Besides, the method developers do not have to make the choice related to the subproblem. Based on the experiments and comparison of different subproblems [18], we have selected subproblems extracted from the STOM, the GUESS and the reference point methods. They all involve reference point information that can be derived from the classification if we know the ranges of the Pareto optimal set. In this case, we set the components of a reference point $\bar{\mathbf{z}}$ as $\bar{z}_i = z_i^\star$ for $i \in I^<$, $\bar{z}_i = \hat{z}_i$ for $i \in I^\leq$, $\bar{z}_i = f_i(\mathbf{x}^c)$ for $i \in I^=$, $\bar{z}_i = \varepsilon_i$ for $i \in I^\geq$ and $\bar{z}_i = z_i^{\text{nad}}$ for $i \in I^\diamond$.

The subproblem coming from the satisficing trade-off method (STOM) [24] has the form:

$$\begin{aligned} &\text{minimize} && \max_{i=1,\ldots,k} \left[\frac{f_i(\mathbf{x}) - z_i^{\star\star}}{\bar{z}_i - z_i^{\star\star}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{\bar{z}_i - z_i^{\star\star}} \\ &\text{subject to} && \mathbf{x} \in S \end{aligned} \tag{2}$$

where the aspiration levels $\bar{z}_i$ must be strictly higher than the corresponding components of the utopian objective vector $z_i^{\star\star}$.

Among the different achievement (scalarizing) functions that can be used in the reference point method [27], we use a basic formulation in the subproblem:

$$\begin{aligned} &\text{minimize} && \max_{i=1,\ldots,k} \left[\frac{f_i(\mathbf{x}) - \bar{z}_i}{z_i^{\text{nad}} - z_i^{\star\star}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - z_i^{\star\star}} \\ &\text{subject to} && \mathbf{x} \in S \end{aligned} \tag{3}$$

Finally, the last subproblem originates from the GUESS method [1]:

$$\begin{aligned} &\text{minimize} && \max_{i \notin I^{\diamond}} \left[\frac{f_i(\mathbf{x}) - z_i^{\text{nad}}}{z_i^{\text{nad}} - \bar{z}_i} \right] + \rho \sum_{i=1}^{k} \frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - \bar{z}_i} \\ &\text{subject to} && \mathbf{x} \in S \end{aligned} \tag{4}$$

The solutions of subproblems (1)-(4) are Pareto optimal [19]. In order to guarantee Pareto optimality, we include in subproblem (4) an augmentation term that was not used in the original formulation. We also need to make some minor changes in the scalarizing function because our reference point originates from classification and we must avoid dividing by zero. Thus, for $i \in I^{\diamond}$, we replace the denominator in the sum term by $z_i^{\text{nad}} - z_i^{\star\star}$. Notice that the aspiration levels $\bar{z}_i$ have to be strictly lower than the components of the nadir objective vector z_i^{nad}. Even though this subproblem relies significantly on the nadir objective vector, which has to be estimated, its success does not heavily depend on the correctness of the estimate (see [18)].

In NIMBUS, the decision maker can also explore a desired number of intermediate solutions between any two solutions. This means that steps of equal length are taken in the decision space between the two selected solutions and the corresponding objective vectors are used as reference points in subproblem (3). In this way, new Pareto optimal solutions are generated. Note that the solutions generated using different subproblems or as intermediate solutions are not all necessarily different [18]. In this case, we only show the different ones to the decision maker.

Next we can formulate the synchronous NIMBUS algorithm. Due to several subproblems and intermediate solutions, the amount of (Pareto optimal) solutions generated increases and, thus, a flexible data management is important. For this reason, we provide a possibility for the decision maker to save solutions in a database. We denote the set of saved solutions by A. At first, we set $A = \emptyset$. The starting point of the solution process can come from the decision maker or it can be some neutral compromise [28] between the objectives. To get the neutral compromise solution we set $\bar{z}_i = (z_i^{\mathrm{nad}} + z_i^\star)/2$ for all $i = 1, \ldots, k$ and solve subproblem (3).

The steps of the NIMBUS algorithm are the following:

1. Generate a Pareto optimal starting point.

2. Ask the decision maker to classify the objective functions at the current solution and to specify aspiration levels and upper bounds (if needed in the classification specified).

3. Ask the decision maker to select the maximum number of different solutions to be generated (between one and four) and solve as many subproblems (among (1)-(4)).

4. Present the different new solutions obtained to the decision maker.

5. If the decision maker wants to save one or more of the new solutions to A, include it/them to A.

6. If the decision maker does not want to see intermediate solutions between any two solutions, go to step 8. Otherwise, ask the decision maker to select the two solutions from among the new solutions or the solutions in A. Ask the number of the intermediate solutions from the decision maker.

7. Generate the desired number of intermediate solutions and project them to the Pareto optimal set. Go to step 4.

8. Ask the decision maker to choose the most preferred one among the new and/or the intermediate solutions or the solutions in A. Denote it as the current solution. If the decision maker wants to continue, go to step 2. Otherwise, stop.

The algorithm is terminated if the decision maker does not want to decrease any objective value or is not willing to let any objective value increase. Otherwise, the search continues iteratively by moving around the Pareto optimal set.

Versatility is the most important characteristic of the NIMBUS algorithm. The decision maker has a variety of strategies available for directing the interactive solution process. Flexibility in the algorithm means that the decision maker is free to change one's mind while (s)he learns more about the problem.

The synchronous NIMBUS method has been used for solving complex problems related to the designing of paper machines [11] and planning paper making processes [6]. The experts who have acted as decision makers have obtained a better picture of the possibilities of the problem without having to input too much preference information.

2. IND-NIMBUS SYSTEM

The IND-NIMBUS system is capable of solving nonlinear multiobjective optimization problems involving even nondifferentiable and nonconvex functions where the variables can be continuous or integer-valued. It can be used in both MS-Windows and Linux operating systems and it has been designed so that it can be connected to different modelling and simulation tools. In this way, it can be used when solving complicated real-world problems that do not have explicit function formulations but function values come, for example, from the solution of a system of partial differential equations.

The basic setting in IND-NIMBUS is the same as in WWW-NIMBUS but the user interface is completely different. The IND-NIMBUS user interface has been built with the wxPython toolset and there exists a high-level NIMBUS GUI framework where different parts of the user interface have been isolated as independent components [25].

After the optimization problem has been specified, the system generates a Pareto optimal neutral compromise solution as a starting point or the solution given by the decision maker is projected onto the Pareto optimal set. This point is the basis of the first classification. This solution is shown to the decision maker as a bar chart where each bar stands for one of the objective functions. The classification of the objective functions can be carried out by indicating desirable values in a bar chart with a mouse. The bars include information about the current objective values with colours as well as the estimated ranges of each objective function in the Pareto optimal set as the end points of the bars. The less colour one can see in the bar, the closer the current value is to the ideal one and, thus, the better it is.

An example of the classification phase is given in Figure 1 with a problem involving five objective functions, where all the others are to be minimized but the third one is to be maximized. The decision maker can classify the functions either

by clicking with the mouse or by specifying desirable values for the objective functions in the boxes next to the bars. If the decision maker wants the function to get as good a value as possible, it is possible to click the big circle next to the ideal value. On the other hand, the function can change freely if it is not classified at all or if the circle next to the nadir value is clicked.

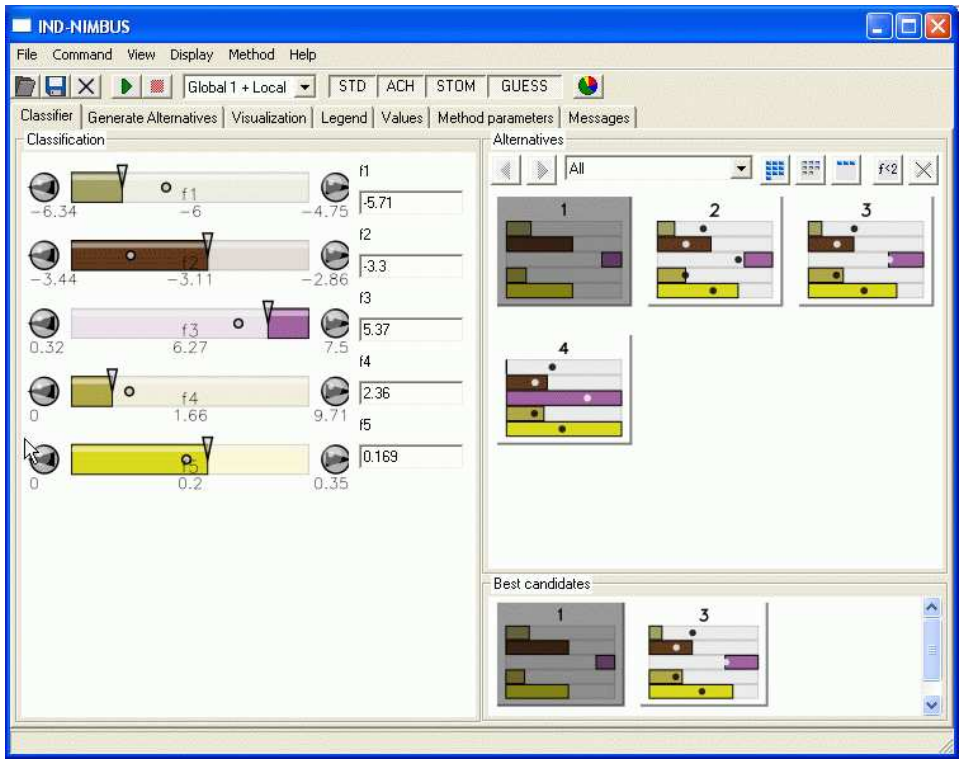

Fig. 1. Classification window

After the classification, the user selects the maximum number (between one and four) of different new solutions to be generated. The system produces them by solving different (single objective) subproblems using some of the underlying optimizers. The user can select the optimizer for each iteration individually. If the user wishes to use a computationally efficient local solver, it is possible to use the proximal bundle method [13]. This method can solve even nondifferentiable problems but it assumes the objective and the constraint functions to be locally Lipschitz continuous and it needs (sub)gradient information.

If the user prefers global optimization, (s)he can select between two variants of an evolutionary algorithm. In this case, the problem to be solved may

contain also integer-valued variables. The two variants use different constraint-handling techniques. One of them is based on adaptive penalties [4] and the other is a method of parameter free penalties [3]. For further details, see [23]. All the optimizers contain technical parameters and the user can change the default values, if necessary.

If the decision maker has decided to use a global solver and (sub)gradient information is available, then the system uses a hybrid solver. This means, that the local solver is started from the final solution of the global solver. Evolutionary algorithms cannot guarantee optimality, but the solution of this hybrid solver is at least locally optimal.

The system shows all the solutions that have been generated during the solution process. An example can be seen in the top right part of Figure 1. Any of them can be hidden if it is not interesting. Furthermore, the decision maker can save any particularly promising or interesting solutions in the set of 'best candidates'. This set can be seen in the bottom right part of Figure 1. In this way, (s)he can comfortably return to previous solutions if they turn out to be interesting. The decision maker can select any of the solutions generated as a starting point of a new classification.

As mentioned in the NIMBUS algorithm, it is also possible to generate a desired number of solutions between any existing solutions. An example of this is given in Figure 2. Here, the decision maker has asked for three new solutions to be generated and they can be seen in the bottom left part of the figure (labelled as output) between the two selected end point solutions (labelled as input in the figure).

The comparison task between any set of solutions is facilitated by using visualizations of the alternatives. The decision maker can compare visually all the solutions generated, the solutions of the previous iteration, the best candidate solutions or a set of any selected solutions. The decision maker can select between bar charts, 3-dimensional bars, value paths, spider-web charts, multiway dot plots, whisker plots and petal diagrams in both absolute and relative scales. It is also possible to filter out some of the solutions by specifying upper or lower bounds or intervals for objective functions. A selection of different visualizations is given in Figure 3.

IND-NIMBUS has been used in solving problems related to process simulation in pulp and paper processes [5; 6] where it has been connected to the BALAS process simulator (http://www.vtt.fi/pro/balas). It has also been used in optimizing paper making in solving problems related to controlling the whole paper machine (as an entireness consisting of many different parts) [12]. In addition, IND-NIMBUS has been connected with the Numerrin software (http://www.numerola.fi/englanti/numerrin02.html) for model based simu-

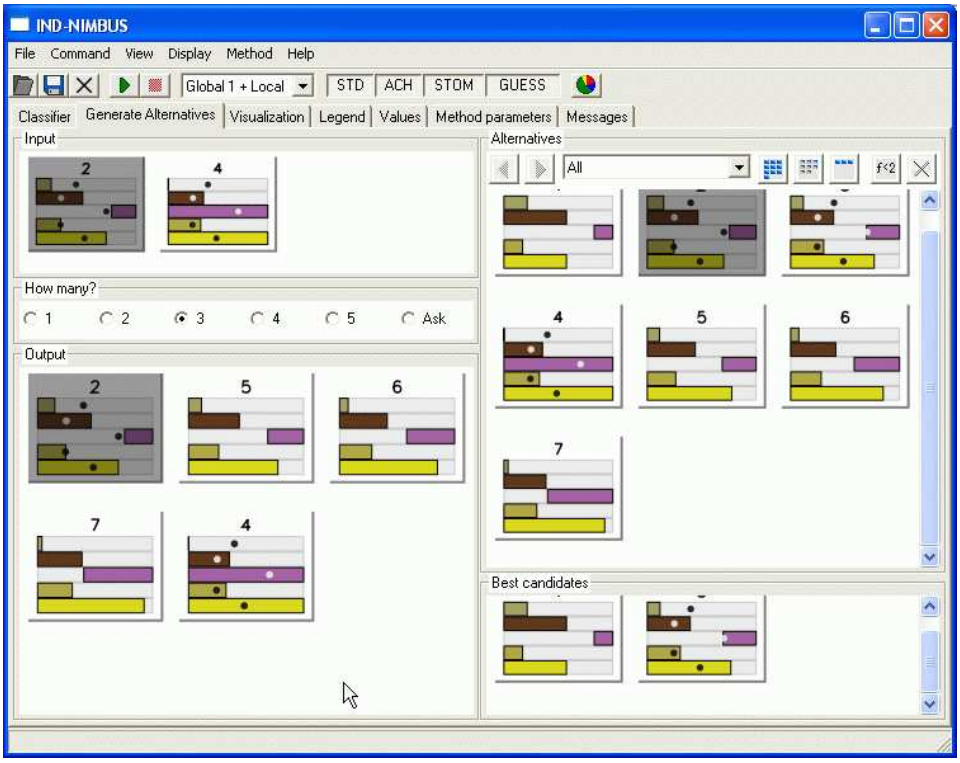

Fig. 2. Intermediate solutions

lation and with this combined tool a design problem related to ultrasonic transducers has been solved [8].

The decision makers involved in the solution processes have found IND-NIMBUS very useful and easy to use. In their opinion, the information exchanged has been understandable and the user interface convenient to use. They have expressed their appreciation to the versatile possibilities of expressing preference information while solving the problem and they have found very satisfactory solutions.

CONCLUSIONS

We have described the main features of a new software system for solving demanding nonlinear multiobjective optimization problems, IND-NIMBUS. It can be connected with different modelling and simulation tools so that problems

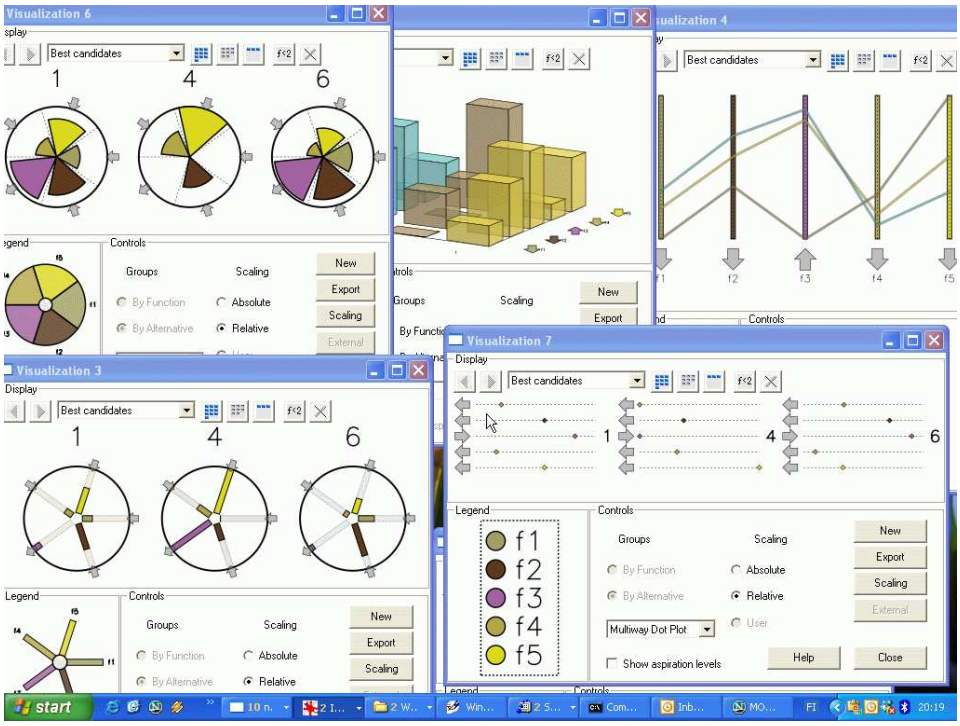

Fig. 3. Different visualizations for three solutions

where function evaluations necessitate, for example, the solution of a system of partial differential equations can be conveniently handled. We have also described the underlying interactive NIMBUS method briefly.

The further development of IND-NIMBUS continues. For example, special emphasis has to be given to the computational efficiency so that solving complex problems will not take too much time. Otherwise, the interactive character of the solution process suffers.

REFERENCES

1. Buchanan J.T.: A Naive Approach for Solving MCDM Problems: The GUESS Method. "Journal of the Operational Research Society" 1997, 48, pp. 202-206.

2. Chankong V., Haimes Y.Y.: Multiobjective Decision Making: Theory and Methodology. Elsevier Science Publishing, New York 1983.

3. Deb K.: An Efficient Constraint Handling Method for Genetic Algorithms. "Computer Methods in Applied Mechanics and Engineering" 2000, 186, pp. 311-338.

4. Hadj-Alouane A.B., Bean J.C.: A Genetic Algorithm for the Multiple-Choice Integer Program. "Operations Research" 1997, 45(1), pp. 92-101.

5. Hakanen J. , Hakala J., Manninen J.: An Integrated Multiobjective Design Tool for Pulp and Paper Process Design. "Applied Thermal Engineering" 2006, 26(13), pp. 1393-1399.

6. Hakanen J., Miettinen K., Mäkelä M.M.: An Application of Multiobjective Optimization to Process Simulation. In: CD-rom Proceedings of ECCOMAS 2004. Eds. P. Neittaanmäki, T. Rossi, S. Korotov, E. Onate, J. Periaux, D. Knörzer. 4th European Congress on Computational Methods in Applied Sciences and Engineering, Vol. II, 2004.

7. Hämäläinen J.P., Miettinen K., Tarvainen P., Toivanen J.: Interactive Solution Approach to a Multiobjective Optimization Problem in Paper Machine Headbox Design. "Journal of Optimization Theory and Applications" 2003, 116(2), pp. 265-281.

8. Heikkola E., Miettinen K., Nieminen P.: Multiobjective Optimization of an Ultrasonic Transducer using NIMBUS. "Ultrasonics" (to appear).

9. Hwang C.L., Masud A.S.M.: Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey. Springer-Verlag, Berlin 1979.

10. Lewandowski A., Wierzbicki A.P. (Eds.). Aspiration Based Decision Support Systems: Theory, Software and Applications. Springer-Verlag, Berlin 1989.

11. Madetoja E.: On Interactive Multiobjective Optimization Related to Paper Quality. Licentiate Thesis, University of Jyväskylä, Department of Mathematical Information Technology, 2003.

12. Madetoja E., Mäkelä M.M., Miettinen K., Tarvainen P.: From Paper Making Simulation and Optimization Towards Engineering Decision Support. Reports of the Department of Mathematical Information Technology, Series B. Scientific Computing, No B3/2005, University of Jyväskylä, Jyväskylä 2005.

13. Mäkelä M.M., Neittaanmäki P.: Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control. World Scientific Publishing Co., Singapore 1992.

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

15. Miettinen K., Mäkelä M.M.: Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: NIMBUS. "Optimization" 1995, 34(3), pp. 231-246.

16. Miettinen K., Mäkelä M.M.: Comparative Evaluation of Some Interactive Reference Point-Based Methods for Multi-Objective Optimisation. "Journal of the Operational Research Society" 1999, 50(9), pp. 949-959.

17. Miettinen K., Mäkelä M.M.: Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet. "Computers & Operations Research" 2000, 27(7-8), pp. 709-723.

18. Miettinen K., Mäkelä M.M.: On Scalarizing Functions in Multiobjective Optimization. "OR Spectrum" 2002, 24(2), pp. 193-213.

19. Miettinen K., Mäkelä M.M.: Synchronous Approach in Interactive Multiobjective Optimization. "European Journal of Operational Research" 2006, 170(3), pp. 909-922.

20. Miettinen K., Mäkelä M.M., Kaario K.: Experiments with Classification-Based Scalarizing Functions in Interactive Multiobjective Optimization. "European Journal of Operational Research" (to appear).

21. Miettinen K., Mäkelä M.M., Mäkinen R.A.E.: Interactive Multiobjective Optimization System NIMBUS Applied to Nonsmooth Structural Design Problems. In: System Modelling and Optimization. Eds. J. Doležal, J. Fidler. Chapman & Hall, London 1996, pp. 379-385.

22. Miettinen K., Mäkelä M.M., Männikkö T.: Optimal Control of Continuous Casting by Nondifferentiable Multiobjective Optimization. "Computational Optimization and Applications" 1998, 11(2), pp. 177-194.

23. Miettinen K., Mäkelä M.M., Toivanen J.: Numerical Comparison of Some Penalty-Based Constraint Handling Techniques in Genetic Algorithms. "Journal of Global Optimization" 2003, 27(4), pp. 427-446.

24. Nakayama J., Sawaragi Y.: Satisficing Trade-Off Method for Multiobjective Programming. In: Interactive Decision Analysis. Eds. M. Grauer, A.P. Wierzbicki. Springer Verlag, Berlin 1984, pp. 113-122.

25. Ojalehto V., Miettinen K., Mäkelä M.M.: Issues of Implementing IND-NIMBUS Software for Interactive Multiobjective Optimization. Reports of the Department of Mathematical Information Technology, Series B., Scientific Computing No B8/2005, University of Jyväskylä, Jyväskylä 2005.

26. Sawaragi Y., Nakayama H., Tanino V.: Theory of Multiobjective Optimization. Academic Press, Inc., 1985.

27. Wierzbicki A.P.: A Mathematical Basis for Satisficing Decision Making. "Mathematical Modelling" 1982, 3, pp. 391-405.

28. Wierzbicki A.P.: Reference Point Approaches. In: Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications. Eds. T. Gal, T.J. Stewart, T. Hanne. Kluwer Academic Publishers, Boston 1999, pp. 9-1–9-39.

29. Wierzbicki A.P., Makowski M., Wessels J. (Eds.). Model-Based Decision Support Methodology with Environmental Applications. Kluwer Academic Publishers, Dordrecht 2000.