

IND-NIMBUS Software for Multiobjective Optimization

VESA OJALEHTO

Department of Mathematical Information Technology
University of Jyväskylä
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä
FINLAND

KAISA MIETTINEN*
and MARKO M. MÄKELÄ†

Abstract: We describe the main features and principles of the IND-NIMBUSdmst system. This system is an implementation of the interactive NIMBUS method for multiobjective optimization. The system can be connected with different types of software describing the mathematical model of the multiobjective optimization problem to be solved.

Keywords: Multiple objective programming, interactive methods, NIMBUS, nonlinear programming, decision support, optimization tool

1 Introduction

Practical optimization applications typically involve several conflicting criteria to be optimized simultaneously. This has been a fruitful starting point for the development of various optimization methods for multiple objectives (see, e.g. [11] and references therein). Despite of that the lack of software for multiobjective optimization is, however, evident. In order to fill this gap WWW-NIMBUS was developed at the University of Jyväskylä in 1995 [14]. The WWW-NIMBUS system is an implementation of the interactive NIMBUS method [11, 12, 13, 14, 16] and it is operating on the Internet being freely available for academic use.

Another feature of practical optimization applications is that the functions characterizing the problem are not necessarily available in an explicit form but they can be evaluated, for example, as a system of partial differential equations. This is a challenge for the optimization software, since it should be able to connect with different simulation codes in a flexible way.

The aim of this research was to develop IND-NIMBUS, a new implementation of the NIMBUS method for industrial optimization applications.

IND-NIMBUS is a portable desktop application targeted for commercial industrial application in addition to academic use.

2 Multiobjective optimization

In classical optimization we have only one objective function to be optimized. However, in real life applications it is typical to have several conflicting objectives, that should be optimized simultaneously. For this reason the classical single objective optimization methods are not applicable as such, but we need special multiobjective optimization methods. For further information see [11].

In multiobjective optimization we consider problems of the form

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S \end{array} \quad (1)$$

involving k (≥ 2) conflicting *objective functions* $f_i : \mathbf{R}^n \rightarrow \mathbf{R}$ that we want to minimize simultaneously. The *decision (variable) vectors* $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ are supposed to belong to the nonempty compact *feasible region* $S \subset \mathbf{R}^n$. *Objective vectors* $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ consist of *objective (function) values* and the image of the feasible

*Helsinki School of Economics, P.O Box 1210, FI-00101 Helsinki, FINLAND

†University of Turku, Department of Mathematics, FI-20014 Turku, FINLAND

region is called a *feasible objective region*.

In multiobjective optimization, objective vectors are regarded as optimal if their components cannot be improved without deterioration to at least one of the other components. More precisely, a decision vector $\mathbf{x}' \in S$ is called *Pareto optimal* if there does not exist another $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ for all $i = 1, \dots, k$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ for at least one index j . Furthermore, an objective vector is Pareto optimal if the corresponding decision vector is Pareto optimal. The set of Pareto optimal objective vectors is called the Pareto optimal set.

Because all the Pareto optimal solutions are mathematically equivalent, we need a *decision maker* (DM) to identify the most preferred one among them. The decision maker is a person who can express preference information related to the conflicting objectives and we assume that less is preferred to more in each objective for her/him.

3 NIMBUS method

NIMBUS is an interactive multiobjective optimization method (see [11, 12, 13, 14, 16]). NIMBUS has been used in solving several real-life problems. Among others, it has been applied in a structural design problem [17], in the optimal control problem of the continuous casting of steel [18] and in the paper machine headbox design [7].

The idea of NIMBUS is that the DM examines the values of the objective functions calculated at a current Pareto optimal decision vector and classifies the objective functions into up to five classes. This means that the DM is asked to indicate (by the means of the classification) what kind of a solution would be more satisfactory than the current one. The classes are for functions whose values

should be decreased,

should be decreased to some aspiration level,

are satisfactory at the moment,

are allowed to increase up till some upper bound,

are allowed to change freely.

In the so-called synchronous NIMBUS method from one to four different single objective subproblems are formulated based on the classification as well as the corresponding aspiration levels

and upper bounds. These subproblems are solved to get one to four new Pareto optimal solutions. (In the earlier versions of the NIMBUS method [11, 12, 13, 14], only one subproblem was formed and solved.) However the same preference information can be the basis of several different subproblems (see [15].)

3.1 Synchronous NIMBUS algorithm

Next we describe shortly the current synchronous NIMBUS method. For more details, see [16]. 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 between the objectives.

1. Get a Pareto optimal starting point.
2. Ask the decision maker to classify the objective functions at the current solution and to specify the possible aspiration levels and upper bounds.
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.
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 solutions in A . Ask the number of new intermediate solutions from the decision maker.
7. Generate the desired number of Pareto optimal intermediate solutions. Go to step 4.
8. Ask the decision maker to choose the most preferred one among the new and/or intermediate solutions or solutions in A . If the decision maker wants to continue, go to step 2. Otherwise, stop.

The algorithm is terminated if the DM 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. In this way, the DM can learn about the problem, adjust ones hopes and finally identify the most desirable solution.

3.2 NIMBUS method implementation

The feasible region S may consist of nonlinear and linear equality and inequality constraints as well as lower and upper bounds for the variables. The objective functions can be either minimized or maximized.

The single objective subproblems that are formed based on the classification information, can be solved with different underlying solvers. If the DM wishes to use a local algorithm, it is possible to use the proximal bundle method [10]. This method can solve even nondifferentiable problems and it assumes the objective and the constraint functions to be locally Lipschitz continuous. Note that it needs (sub)gradient information.

Alternatively, if the DM wishes to use a global optimization solver, he can solve the problem with an genetic algorithm [3]. NIMBUS contains two variants of genetic algorithms with different constraint-handling techniques, based on adaptive penalties [4] and method of parameter free penalties [2]. See also [19]

It is also possible to use other global solvers like Controlled Random Search (CRS) [20] or Differential Evolution [22] global solvers, but at this time these are available only for internal use. These solvers have two different variants, due their constraint-handling techniques.

3.3 WWW-NIMBUS

WWW-NIMBUS is an freely available implementation of the NIMBUS algorithm operating on the Internet [14]. WWW-NIMBUS has been developed for solving nonlinear and mathematically challenging multiobjective optimization problems. WWW-NIMBUS can be accessed at <http://nimbus.mit.jyu.fi/> and can be used, for example, for teaching purposes.

The problem to be solved in WWW-NIMBUS can be specified for the system either by filling a web form or as a subroutine. The classification of the objective functions can be carried out symboli-

cally or graphically by indicating desirable changes in a bar chart with a mouse.

In the WWW-NIMBUS system the user can save, load and modify problems. The flexible database management enables saving selected solutions whenever required. In this way, the user can comfortably return to previous solutions if they turn out to be interesting, after all. The user can also graphically illustrate a (sub)set of saved solutions and not only the latest ones, and generate intermediate solutions between any two solutions. Because of the structure of the algorithm, all the solutions handled are Pareto optimal.

4 IND-NIMBUS

IND-NIMBUS is a multi-platform desktop application implementing the NIMBUS algorithm. Unlike WWW-NIMBUS, it does not contain any tools to formulate the optimization problem. Instead, it is assumed that the problem is formulated in another program, which provides the necessary information for the IND-NIMBUS software.

In IND-NIMBUS, the DM has access to all solutions generated during the process. Any solution can be selected as a starting solution for a new classification, or as an end point for generating intermediate solutions. In this way, the DM can generate a representation of the Pareto optimal set according to her/his preferences for further study

In order to organize these solutions, the DM can create different filters, for example to show only those solutions where the first objective function has values above zero. There is also a possibility to view only those solutions obtained from the last classification or alternative generation. The DM can also store the best solution candidates to the "Best Candidates" panel at any point during the solution process. The undesirable solutions can be deleted from the solution list.

The graphics used in IND-NIMBUS are generated using a standalone package, which can also be used to generate graphics for other NIMBUS implementations. One of the main duties of the graphics package is to generate the visualization graphics for comparing different Pareto optimal solutions. These visualization graphics include two-dimension and three-dimension bar-chart, spider web chart, value paths, whisker plot, petal diagram and multi-way dot plot. The same filtering tools are available for visualizations as for other action windows.

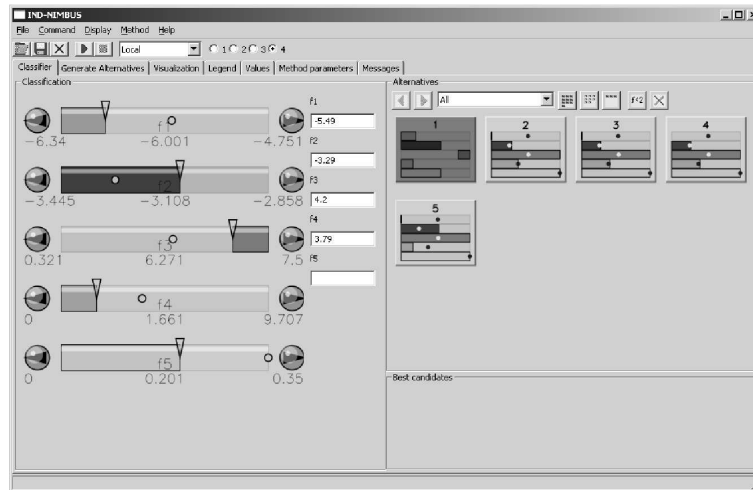


Figure 1: IND-NIMBUS classification window

Numerical variable and objective function values of each solution can also be viewed, and exported, from the IND-NIMBUS software. There also exists views for changing the default parameter values of optimization solvers used, and for viewing possible warning messages these solvers have produced during the optimization process.

The IND-NIMBUS classification window can be seen in Figure 1. Here the classification information is given by clicking the objective function bars on the left. The DM can give the preferred aspiration level or the upper bound to the edit box next to each function bar. Alternatives between any two Pareto optimal solutions can be generated in a secondary window, not shown here.

It should be noted that the IND-NIMBUS user interface is concurrent. In other words, the user can move between different windows, even if some operation is still running. Nevertheless, at this time there can be only one concurrent computational operation running at a time.

4.1 Using IND-NIMBUS with third party applications

As stated previously, the IND-NIMBUS software does not include tools to formulate optimization problem to be solved. Therefore, IND-NIMBUS software must be used in conjunction with a external simulation software. The problem dimensions, objective functions, variable names, and other information defining the problem in question must be specified in a external software. For this purpose, the following applications have been used as IND-NIMBUS test cases.

BALAS[®] [1, 5, 6] is a process simulator developed at the VTT – Technical Research Center of Finland. It operates on the Windows operating system, consisting of a separate program Flow-sheet for the creation of the process flowsheet and the BALAS software, which formulates the process simulation problem to be optimized.

Numerrin 2.0 is a Linux based software for phenomenon based physical modeling of real-world systems. Numerrin models are created by the Fortran 95 programming language, using the finite element analysis library included [8].

The virtual paper making simulation and optimization system (MOP) is designed to provide support for solving complex paper making problems in a paper machine. It contains several unit process models that are combined as one large multidisciplinary simulation model [9]. The MOP software operates on the Windows platform.

MATLAB[®] is a interactive environment that can be used to solve a wide range of problems. MATLAB is used with IND-NIMBUS system to develop a multicriteria model for intensity modulated radiotherapy [21].

In practice, there exist two different ways of connecting the optimization problem to the IND-NIMBUS system. In a simpler approach, the problem can be given as an executable file. This executable contains or dynamically links to the IND-NIMBUS software for the actual optimization. Furthermore, the executable must be able to access all tools needed to formulate and solve the problem in question. During the optimization process this executable is run for each of the NIMBUS opera-

tions as a separate process, and the communication between the executable and the IND-NIMBUS software is done by services provided by the underlying operating system.

However, it is likely that the simulation processes generated by a simulation software cannot be separated to a single executable. In this case, the simulation process and the IND-NIMBUS software are controlled with a separate application, which contains the NIMBUS implementation.

In both approaches the communication between IND-NIMBUS and the problem entity is done via standard files. There exist some basic facilities for socket based communication, but these have not yet been used in any of the test cases used in the IND-NIMBUS development.

As mentioned earlier, the IND-NIMBUS software has some data visualization tools but it is also possible to use problem domain specific tools related to each industrial application to examine different solutions generated during the optimization process. For this purpose, the problem designer can specify a set of files that the problem generates during the objective function calculations. These files are stored with the objective function values, and they can be exported to the industrial application for further referencing.

5 Conclusions and Future Plans

The IND-NIMBUS software is a desktop application that has been successfully connected with several different applications to solve multiobjective optimization problems. During this process, user feedback from different test cases has given new ideas how to improve the IND-NIMBUS software. Some of these features and ideas are discussed here.

During the IND-NIMBUS development it was apparent that the NIMBUS Fortran77 implementation cannot be easily extended to meet with changing requirements. One of the most problematic aspect of the current implementation is the lack of control and communication during the optimization process. Possibilities to affect the running optimization process in a controlled way are rather limited, or non existent. Furthermore, during the optimization the Fortran77 implementation provides little information about the course of the process, so the DM is forced to either wait for the final result or to kill the process when it has taken too long time in his/her opinion.

In addition, several parts of the NIMBUS algorithm could be run in parallel. Using suitable tools, simple parallelization of the initial step and classification operations should not be an overwhelming task. As the current CPU development trends are heading towards multiprocessor systems, the parallelization could in the future offer clear advantages regarding the time used for the optimization process, even in standard desktop environments. Furthermore, the industrial applications are often computationally very demanding, so with suitable parallelization implementation the optimization process could be distributed to several different computers.

It could also be of some interest to store the different objective function values obtained during the optimization process. This information could be used to examine the simulation model after the optimization, and with these values it would be possible to eliminate duplicate objective function calculations for the same, or similar, decision variable values.

The Pareto optimal solution set obtained during the optimization process could be better examined with more advanced filtering tools. Furthermore, as the solutions are generated in a treelike structure, this structure could be shown to the DM to give him/her a better view of the whole solution process.

Acknowledgments

The IND-NIMBUS development has been a part of the NIMBUS and MASIT01 projects supported by TEKES – Finnish Funding Agency for Technology and Innovation. MSc. Paavo Nieminen, Ph.Lic. Jussi Hakanen and Ph.Lic. Elina Madetoja have given valuable help in connecting IND-NIMBUS with different industrial applications. MSc. Tommi Ronkainen has been the principal developer of IND-NIMBUS software.

References

- [1] Balas, Process Simulation Software. VTT Processes, <http://www.vtt.fi/pro/balas/>, 2006.
- [2] K. Deb. *An Efficient Constraint Handling Method for Genetic Algorithms*. Computer Methods in Applied Mechanics and Engineering, Vol. 186, pp. 11 – 338, 2000.

- [3] G.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc., 1989.
- [4] A.B. Hadj-Alouane and J.C. Bean. *A Genetic Algorithm for the Multiple-Choice Integer Program*. *Operations Research*, Vol. 45, No. 1, pp. 92–101, 1997.
- [5] J. Hakanen, J. Hakala, and J. Manninen. *An integrated multiobjective design tool for process design*. *Applied Thermal Engineering*, Vol. 26, No. 13, pp. 1993–1399, 2006.
- [6] J. Hakanen, Y. Kawajiri, K. Miettinen, and L.T. Biegler. *Interactive multi-objective optimization of Simulated Moving Bed processes using IND-NIMBUS and IPOPT*. Working Papers W-404, Helsinki School of Economics, 2006.
- [7] J.H. Hämäläinen, K. Miettinen, P. Tarvainen, and J. Toivanen. *Interactive Solution Approach to a Multiobjective Optimization Problem in Paper Machine Headbox Design*. *Journal of Optimization Theory and Applications*, Vol. 116, pp. 265–281, 2003.
- [8] E. Heikkola, K. Miettinen, and P. Nieminen. *Multiobjective Optimization of an Ultrasonic Transducer using NIMBUS*. *Ultrasonics* (to appear).
- [9] E. Madetoja. *On Interactive Multiobjective Optimization Related to Paper Quality*. Licentiate thesis, Department of Mathematical Information Technology, University of Jyväskylä, 2003.
- [10] M.M. Mäkelä and P. Neittaanmäki. *Nonsmooth Optimization Analysis and Algorithms with Applications to Optimal Control*. World Scientific, Singapore, 1992.
- [11] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [12] K. Miettinen and M.M. Mäkelä. *Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: NIMBUS*. *Optimization*, Vol. 34, pp. 231–246, 1995.
- [13] K. Miettinen and M.M. Mäkelä. *Comparative Evaluation of Some Interactive Reference Point-based Methods for Multi-Objective Optimization*. *Journal of the Operational Research Society*, Vol. 50, pp. 949–959, 1999.
- [14] K. Miettinen and M.M. Mäkelä. *Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet*. *Computers & Operations Research*, Vol. 27, pp. 709–723, 2000.
- [15] K. Miettinen and M.M. Mäkelä. *On Scalarizing Functions in Multiobjective Optimization*. *OR Spectrum*, Vol. 24, No. 2, pp. 193–213, 2002.
- [16] K. Miettinen and M.M. Mäkelä. *Synchronous approach in interactive multiobjective optimization*. *European Journal of Operational Research*, Vol. 170, No. 3, pp. 909–922, 2006.
- [17] K. Miettinen, M.M. Mäkelä, and R.A.E. Mäkinen. *Interactive Multiobjective Optimization System NIMBUS Applied to Nonsmooth Structural Design Problems*. *System Modelling and Optimization*, Proceedings of the 17th IFIP Conference on System Modelling and Optimization, pp. 379–385. Chapman & Hall, London, 1996.
- [18] K. Miettinen, M.M. Mäkelä, and T. Männikkö. *Optimal Control of Continuous Casting by Nondifferentiable Multiobjective Optimization*. *Computational Optimization and Applications*, Vol. 11, pp. 177–194, 1998.
- [19] K. Miettinen, M.M. Mäkelä, and J. Toivanen. *Numerical Comparison of Some Penalty-Based Constraint Handling Techniques in Genetic Algorithms*. *Journal of Global Optimization*, Vol. 27, No. 4, pp. 427–446, 2003.
- [20] W.L. Price. *Global optimization by Controlled Random Search*. *Journal of Optimization Theory and Applications*, Vol. 40, No. 3, pp. 333–348, 1983.
- [21] H. Ruotsalainen, K. Miettinen, E. Boman, and J. Hämäläinen. *Multicriteria Optimization for Intensity Modulated Radiotherapy Based on Decision Maker's Knowledge*. manuscript available via private communication: Henri.Ruotsalainen@uku.fi.
- [22] R. Storn and K. Price. *Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*. *Journal of Global Optimization*, Vol. 11, No. 4, pp. 341–359, 1997.