Working Paper

SAP – Modular Tool for Specification and Analysis of User Preferences in Multiple-Criteria Model Analysis

Janusz Granat, Marek Makowski

WP-95-73
December 1995

IIASA
International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

SAP – Modular Tool for Specification and Analysis of User Preferences in Multiple-Criteria Model Analysis

Janusz Granat, Marek Makowski

WP-95-73
December 1995

IIASA International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

Foreword

An important part of decision support is the process of decision selection. This process can be supported by Multiple-Criteria Decision Analysis (MCDA). Most methods for MCDA are based on interactions with the users who are supposed to state preferences with respect to a selection of an efficient solution. The present paper treats the problem that such preferences are not always hard by allowing to use a fuzzy specification.

A modular software tool which supports this approach has been developed within the collaboration between the Methodology of Decision Analysis Project and the Institute of Control and Computation Engineering, Warsaw University of Technology. The tool has been applied at IIASA to two models examined in collaboration with the Water Resources and Land Use and Land Cover Change projects at IIASA. It has been also applied at the Warsaw University of Technology to several engineering applications. The presented paper documents the tool and provides a tutorial for its use.

Abstract

Model based Decision Support Systems (DSS) often use multiple-criteria optimization for selecting Pareto-efficient solutions. Such a selection is based on interactive specification of user preferences. This can be done by specification of aspiration and reservation levels for criteria. Diverse graphical user interface could be used for specification of these levels as well as for interpretation of results. In the approach presented in this paper the specified aspiration and reservation levels are used for generation of component achievement functions for corresponding criteria. Such functions can be interpreted as fuzzy membership functions or as functions, which reflect the degree of satisfaction with given values of criteria.

The paper outlines the methodological background and modular structure of a DSS shell for multiple-criteria analysis of decision problems that can be represented as Linear Programming (LP) or Mixed Integer Programming (MIP) problems. The DSS shell has been used at IIASA for analysis of decision problems in water quality management and land use for sustainable development planning. The pilot implementation of one component of that DSS, namely the modular software tool for interactive specification of user preferences is described in more detail. The tool has been also used as in a DSS for analysis of non-linear problems in several engineering applications.

Keywords: Multiple-Criteria Optimization, Decision Support Systems, Interactive Specification of Preferences, Aspiration-Reservation Based Decision Support, Fuzzy Sets, Linear Programming, Mixed Integer Programming.

Contents

SAP – Modular Tool for Specification and Analysis of User Preferences in Multiple-Criteria Model Analysis

Janusz Granat, Marek Makowski*

1 Introduction

A Decision Support System (DSS) is a computerized tool which helps to analyze a decision problem. For any model-based DSS one can distinguish the following three groups of related modeling activities, underlying methodologies and software:

Model generation: Generation of a *core model* which is a representation in terms of mathematical programming (however, without specification of goal functions) of all logical and physical relations between variables representing the decision problem being examined. The core model implicitly defines a set of feasible solutions but it does not contain any preferential structure of a Decision Maker (DM), which is specified and later modified during the analysis of the model.

Model analysis: Adding to the core model a representation of a preferential structure of a DM. In other words the user selects from the set of all feasible solutions (defined by the *core model*) a subset of solutions that are acceptable and then he/she provides information that is used for further selection from such a subset of one solution that corresponds best to the preferences of the DM. This can be done by the selection of criteria for a multiple-criteria based model analysis, or by the selection of one criterion and additional constraints for a single-criterion optimization. Each method of the representation of a preferential structure for a user has a number of parameters that have to be set by the user in order to formulate an optimization (or simulation) problem. The analysis is often done in an interactive way, thus allowing a user analysis of previously obtained solutions, changing the representation of his/her preferential structure thus formulating a corresponding underlying optimization problem.

Problem solving: Solving of a corresponding mathematical programming problem requires a robust and efficient solver that can handle a corresponding type of optimization or simulation problem in a way that is transparent for a user of a DSS.

This paper deals with an extension of the aspiration-led multiple-criteria optimization based model analysis, which is commonly called Aspiration-Reservation Based Decision Support (ARBDS). Today, ARBDS is one of the most promising techniques for model

*Institute of Control and Computing Engineering, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland.

analysis for decision support. However, one of the major constraints for wide applications of any method that requires interaction with the user is the lack of modular software tools that can be used for an implementation of a problem specific DSS. Therefore the modular tool, called SAP, which facilitates the interaction with the user by providing all the functions necessary for interactive analysis of a problem using the ARBDS methodology has been developed and is documented by this paper. The name SAP is an abbreviation of the **S**pecification and **A**nalysis of **P**references, named after the methodological background outlined in Section 2.

The remaining part of this paper is organized in the following way. Section 2 provides an outline of the ARBDS methodology. Section 3 summarizes the structure of a DSS shell which can be used for analysis of decision problems represented by an LP problem and presents functions of SAP. Section 4 provides assumptions adopted for SAP and the information needed for using SAP by both a user of a ready application and by a developer who wants to include SAP into his/her application. Section 5 informs about the availability of SAP. Appendix A contains a tutorial session which illustrates the use of SAP within the DSS shell, which is available free of charge for non-commercial research and educational purposes.

2 Methodological background

Discussion on different approaches to decision support is clearly beyond the scope of this paper. A large bibliography can be found e.g. in [EKO90, KLW91, LeW89a, Mak94a, Ste92, WeW93]. We will deal with one of the most successful (see e.g. [KoW89] for a justification of this statement) class of DSS, namely with model based DSS which use aspiration-led multiple objective optimization as a tool for computing and selecting efficient solutions. This approach, originally proposed in [Wie80, Wie82], now has more than a dozen slightly different methodological versions. The theoretical and methodological backgrounds for aspiration based decision analysis and support is provided e.g. in [Wie80, LeW89b]. A unified procedure that covers most of those approaches has been proposed in [GaS94a, GaS94b].

An extension of the aspiration-led multiple criteria model analysis is called Aspiration-Reservation Based Decision Support (ARBDS). The ARBDS methodology has been implemented in a number of DSS presented in [LeW89a]. The relations between ARBDS and other approaches to multiple-criteria optimization are discussed in more detail in [Mak94c]. ARBDS can be also considered (see [OgL92]) as an extension of Goal Programming (see e.g. [ChC67] for details), most probably the oldest technique for multiple-criteria analysis of linear programs. Today, ARBDS is one of the most promising techniques for model based decision support.

Here we summarize the ARBDS method as a two-stage approach:

- First, a *core model* is specified and generated. The core model contains only a set of constraints that correspond to logical and physical relations between the variables used in the model. Those variables should also include variables that represent potential criteria (goals, performance indices). In the preparatory stage a DM selects (from the core model variables) a set of criteria that will be used for the analysis of the model, and specifies a type for each criterion. The selected type declares that a criterion is either minimized or maximized or targeted at a given value (`goal` type of a criterion, see Section 4.2.1 for details)[1]. After the selection of a set of criteria, LP-Multi automatically

[1] Note, that a variable can represent also more complicated forms of criteria (like following a trajectory,

performs a series of optimizations in order to compute the Utopia point and an approximation of the Nadir point[2]. The preparatory stage is finished with computation of the so-called *compromise solution* which corresponds to a problem for which the aspiration and reservation levels are (automatically) set to the Utopia and an approximation of the Nadir points, respectively.

- Second, an interactive procedure is used for helping the user in selecting an efficient solution that best corresponds to his/her preferences. During such a procedure a DM specifies goals and preferences, including values of criteria that he/she wants to achieve and to avoid. The vectors composed of those values are called *aspiration* and *reservation* levels, respectively. Such a specification defines component achievement functions (see Section 2.1) which are used for selection of a Pareto optimal solution. Such a solution is achieved by generation of additional constraints and variables, which are added by LP-Multi to the *core model* thus forming an optimization problem, whose solution results in a Pareto solution that is nearest (in the sense of a measure defined by the aspiration and reservation levels) to the specified aspiration levels (or uniformly better than these levels, if they are attainable).

The SAP handles the interaction with the user in the second stage of the problem analysis, therefore we will provide more details about this stage, which can be described in the form of the following steps:

1. The DM specifies new aspiration and reservation levels for all criteria which have the default (see Section 2.3) status. For each stabilized criterion (if any), the DM specifies a corresponding target (desired) value and aspiration and reservation levels for a deviation from the specified target value. The details of this option are provided in Section 4.3.1. Optionally, the DM can specify for those criteria his preferences in terms of fuzzy sets. The methodological background for this option is presented in Section 2.2 and its implementation is documented in Section 4.3.2.
2. The DM can change the status of each criterion. The default status can be changed to *stabilized*, *inactive* or *disregarded.* This is supported by the *Status* option (see Section 4.3.3).
3. The DM can analyze criteria values of the solutions computed so far (together with values of aspiration and reservation levels used for each solution). This part of analysis is supported by the *History* option (see Section 4.3.5).
4. The DM may want to store a currently analyzed solution of the underlying LP or MIP problem for a more detailed analysis (which is typically problem specific). This can be done by a selection of *Store* submenu from the *Solution* menu of LP-Multi (see Figure 2).
5. The DM can freely switch between the actions summarized above until he/she decides that his/her preferences are properly represented for the next optimization, which is selected as described in Section 4.3.6. Once the optimization is selected, the LP-Multi takes control of the program flow, LP-Multi generates a single-criterion optimization problem whose solution is a Pareto-efficient solution which corresponds to the current

minimization of a distance, etc.). Examples of different types of criteria (which are formally represented by a variable, whose value is either minimized or maximized) and the way to handle so-called soft constraints in the framework of ARBDS can be found e.g. in [Mak94c].

[2]Utopia and Nadir points (in the space of criteria) are vectors composed of best and worst values of the criteria in the efficient set. It can be shown (c.f. e.g. [IsS87]) that a computation of a Nadir point for problems with more than two criteria may be very difficult. In our approach the Nadir point plays a minor informative role (it only bounds values of corresponding reservation levels). Therefore there is no justification for spending resources in order to get a better approximation. Hence, we assume as an approximation of Nadir the worst value (obtained during the analysis) of a corresponding criterion.

preference structure of the DM (see Section 2.1 for details) and executes an appropriate solver, which computes such a solution. The DM regains control of the program when the solution of the last specified problem is ready and added to the previously obtained solutions.

The steps described above are repeated in order to explore various Pareto-efficient solutions, until a satisfactory solution is found or until the user decides to break the analysis. In either case the analysis can be continued from the last obtained solution at a later time.

2.1 Selection of a Pareto-optimal solution

Multiple-criteria optimization methods typically assume that a multi-objective problem is converted into an auxiliary parametric single-objective problem whose solution provides a Pareto-optimal point[3]. Different methods apply different conversions but most commonly known methods can be interpreted (see [Mak94c]) in the terms of Achievement Scalarizing Function (ASF). The concept of ASF has been introduced by Wierzbicki (see e.g. [Wie77, Wie86, Wie92] for the mathematical foundations, interpretations and applications) and it is very useful for comparing different approaches to multiple-criteria optimization.

The selection of the Pareto-optimal point depends on the definition of the ASF, which – for the aspiration-led model analysis – also includes a selected aspiration point. Most of those methods use the maximization of an ASF in the form:

$$s(q,\bar{q},w) = \min_{1 \le i \le n} \{w_i(q_i - \bar{q}_i)\} + \epsilon \sum_{i=1}^{n} w_i(q_i - \bar{q}_i) \tag{1}$$

where $q(x) \in R^n$ is a vector of criteria, $x \in X_0$ are variables defined by the core model, X_0 is set of feasible solutions implicitly defined by the core model, $\bar{q} \in R^n$ is an aspiration point, $w_i > 0$ are scaling coefficients and ϵ is a given small positive number. Maximization of (1) for $x \in X_0$ generates a properly efficient solution with the trade-off coefficients (as recomputed in terms of u_i) smaller than $(1 + 1/\epsilon)$. For a non-attainable $\bar{q}$ the resulting Pareto-solution is the nearest (in the sense of a Chebyshev weighted norm) to the specified aspiration level $\bar{q}$. If $\bar{q}$ is attainable, then the Pareto-point is uniformly better. Setting a value of ϵ is itself a trade-off between getting a too restricted set of properly Pareto solutions or a too wide set practically equivalent to weakly Pareto optimal solutions. Assuming the ϵ parameter to be of a technical nature, the selection of efficient solutions is controlled by the two vector parameters: $\bar{q}$ and w.

There is a common agreement that the aspiration point is a very good controlling parameter for examining a Pareto set. Much less attention is given to the problem of defining the weighting[4] coefficients w. A detailed discussion on weights in a scalarizing function is beyond the scope of this paper. The four commonly used approaches are summarized in [Mak94c]. In practical applications the most promising approach is based on calculation of weights (that are used in definition of Chebyshev norm mentioned above) with help of the aspiration level $\bar{q}$ and a reservation level $\underline{q}$ (the latter is composed of values of criteria that the user wants to avoid). This is the ARBDS approach that has been introduced by the DIDAS family (described in [LeW89a]) of DSS.

[3] A solution is called Pareto-optimal (or efficient) solution, if there is no other solution for which at least one criterion has a better value while values of remaining criteria are the same or better. In other words, one can not improve any criterion without deteriorating a value of at least one other criterion.

[4] Note that the weights w should not (see e.g. [Mak94c, Nak94]) be used for conversion of a multiple criteria problem into a into a single criterion problem with a weighted sum of criteria. In the function (1) they play a different role than in a weighted sum of criteria.

The ASF for the ARBDS approach usually takes the form:

$$\mathcal{S}(q,\bar{q},\underline{q}) = \min_{1\le i\le n} u_i(q_i,\bar{q}_i,\underline{q}_i) + \epsilon\sum_{i=1}^{n} u_i(q_i,\bar{q}_i,\underline{q}_i) \qquad (2)$$

where $\bar{q}, \underline{q}$ are vectors (composed of $\bar{q}_i, \underline{q_i}$, respectively) of aspiration and reservation levels respectively, and $u_i(q_i,\bar{q}_i,\underline{q}_i)$ are the corresponding component achievement functions (defined later in detail), which can be simply interpreted as nonlinear monotone transformations of q_i taking into account the information contained in $\bar{q}_i$ and $\underline{q}_i$. Maximization of the function (2) over the set of feasible solutions X_0 defined by the corresponding core model provides a properly Pareto-optimal solution with the properties discussed above for the function (1).

The ASF implemented in LP-Multi is a modification of the function (2). The modification has been stimulated by some applications for which it is often useful to temporarily disregard some of the criteria. A criterion for which the user does not wish to define the corresponding component scalarizing function is called in LP-Multi *an inactive criterion* (see Section 4.3.3). Inactive criteria are also useful for computing a good approximation of a nadir point. However, completely disregarding a criterion from the ASF may result in both numerical problems (caused by a degenerated problem) and in a random value of the criterion (which may be unnecessarily bad and can in turn result in a bad approximation of a nadir point, see [Mak94c] for more details). Therefore, the following form of the ASF is implemented in LP-Multi in order to facilitate a proper handling of inactive criteria:

$$\mathcal{S}(q,\bar{q},\underline{q}) = \min_{i\in I} u_i(q_i,\bar{q}_i,\underline{q}_i) + \epsilon\sum_{i\in I} u_i(q_i,\bar{q}_i,\underline{q}_i) + \epsilon\sum_{i\in\bar{I}} s_i q_i \qquad (3)$$

where I and $\bar{I}$ are sets of indices of active and inactive criteria, respectively, and the scaling coefficients s_i are defined by:

$$s_i = \frac{\text{sign}(q_i^U - q_i^N)}{\max(\epsilon, |q_i^U - q_i^N|)} \qquad (4)$$

where `sign(x)` is a function that returns 1 for non-negative numbers and -1 otherwise, and q_i^U and q_i^N are utopia and approximation of nadir values, respectively. One can easily show that the treatment of a criterion as an *inactive* one has a similar effect to selecting the corresponding aspiration level close to the approximation of Nadir for that criterion. Note, that for all criteria being active the ASF defined by (3) is equivalent to that of (2).

Component achievement functions $u_i(\cdot)$ are strictly monotone (decreasing for minimized and increasing for maximized criteria, respectively) functions of the objective vector component q_i with values

$$u_i(q_i^U,\cdot) = 1+\bar{\gamma}, \qquad u_i(\bar{q}_i,\cdot) = 1, \qquad u_i(\underline{q}_i,\cdot) = 0, \qquad u_i(q_i^N,\cdot) = -\underline{\gamma} \qquad (5)$$

where $\bar{\gamma}$ and $\underline{\gamma}$, are given positive constants, typically equal to 0.1 and 10, respectively.

The piece–wise linear component achievement functions u_i proposed in [Wie86] are defined by (6) and by (7) for minimized and maximized criteria, respectively.

$$u_i(q,\bar{q},\underline{q}) = \begin{cases} \zeta_i w_i(\bar{q}_i - q_i) + 1, & \text{if} \quad q_i < \bar{q}_i \\ w_i(\bar{q}_i - q_i) + 1, & \text{if} \quad \bar{q}_i \le q_i \le \underline{q}_i \\ \eta_i w_i(\underline{q}_i - q_i) & \text{if} \quad \underline{q}_i < q_i \end{cases} \qquad (6)$$

$$u_i(q,\bar{q},\underline{q}) = \begin{cases} \zeta_i w_i(\underline{q}_i - q_i) & \text{if} \quad q_i < \underline{q}_i \\ w_i(\bar{q}_i - q_i) + 1, & \text{if} \quad \underline{q}_i \le q_i \le \bar{q}_i \\ \eta_i w_i(\bar{q}_i - q_i) + 1, & \text{if} \quad \bar{q}_i < q_i \end{cases} \qquad (7)$$

where $w_i = 1/(\underline{q}_i - \bar{q}_i)$, and ζ_i, η_i $(i = 1, 2, \ldots, n)$ are given parameters, which are set in such a way that u_i takes the values defined by (5).

However, in order to allow for either specification of only aspiration and reservations levels or for additional specification of preferences (for the criteria values between aspiration and reservation levels) in terms of fuzzy sets (as described in Section 2.2) the SAP supports specification of the component achievement functions in a more general form than that of eq. (6,7). Therefore, the piece-wise linear functions u_i are defined by segments u_{ji}:

$$u_{ji} = \alpha_{ji} q_i + \beta_{ji}, \qquad q_{ji} \leq q_i \leq q_{j+1,i} \qquad j = 1, \ldots, p_i \tag{8}$$

where p_i is a number of segments for i-th criterion. Practical applications shows that sometimes it is useful to set $\bar{q}_i = q_i^U$ and/or $\underline{q}_i = q_i^N$. Therefore, in order to handle also component achievement functions composed of only one segment (in cases when an aspiration level is set to the Utopia value and a reservation level is equal to an approximation of Nadir) SAP allows for $p_i \geq 1$.

The coefficients defining the segments are given by:

$$\alpha_{ji} = \frac{u_{j+1,i} - u_{ji}}{q_{j+1,i} - q_{ji}} \tag{9}$$

$$\beta_{ji} = u_{ji} - \alpha_{ji} q_{ji} \tag{10}$$

where points (u_{ji}, q_{ij}) are interactively defined with the help of SAP (see Section 4.3.1 for details). Concavity of the piece-wise linear functions $u_i(q_i)$ defined by segments (8) can be assured by a condition:

$$\alpha_{1i} > \alpha_{2i} > \ldots > \alpha_{p_i i} \qquad j = 1, \ldots, p_i \tag{11}$$

Note that the component achievement functions u_i defined by (8) take the same form for minimized and maximized criteria. However, one should add (in addition to the condition (11) that assures concavity) a condition:

$$\alpha_{1i} < 0 \qquad i \in I^{\min} \tag{12}$$

$$\alpha_{p_i i} > 0 \qquad i \in I^{\max} \tag{13}$$

where $I^{\min}$ and $I^{\max}$ are sets of indices of criteria minimized and maximized, respectively. The conditions (12,13) are fulfilled automatically for the component achievement functions u_i specified with the help of SAP.

2.2 Interactive specification of preferences

Various graphical user interfaces can be used for specification of aspiration and reservation levels as well as interpretation of solutions. In the approach presented in this paper the specified aspiration and reservation levels are used for generation of the component achievement functions (6,7). Such functions – for the parameters ζ_i and η_i set to zero – can be interpreted in the terms of the fuzzy membership functions (MF discussed in detail by Zimmermann in [Zim78, Zim85]) as functions, which reflect the degree of satisfaction with a given set of criteria values. The graphs of these functions are presented to the user on the screen. Such graphical presentation allow not only specification of the user preferences but also helps him/her in interpretation of the solutions. This analysis can be done by projections of multidimensional criteria space into two dimensional spaces composed for each criterion of its values and the degree of satisfaction of meeting preferences expressed by aspiration and reservation levels.

Theoretical background and a number of applications of fuzzy sets to decision making are discussed in [Zim87, Sak93]. These approaches assume that the MF is elicited before the interactive analysis of the problem. Interactive fuzzy multi-objective programming as proposed in [SeS88, Sak93] uses a given set MF (one MF for each criterion) for the interactive procedure in which the user specifies the aspiration levels of achievement of the membership values for all of the membership functions, called the reference membership level. Therefore in this approach the user can not change aspiration and reservation levels in terms of criteria values, because they have to be specified a priori for the definition of MF. In the ARBDS approach the MF is not elicited at an initial iteration but the user is allowed to interactively change it upon analysis of obtained solutions. So we are allowing the change of the membership function due to learning process. The definition of linear MF is done by specification of two points, which are equivalent to specification of aspiration and reservation levels in the criteria space.

The ARBDS approach uses so called extended-valued membership function. Such an extension of the membership function concept has been proposed by Granat and Wierzbicki in [GrW94], who also suggested a method of constructing various forms of order-consistent component achievement scalarizing functions based on membership functions describing the satisfaction of the user with the attainment of separate objectives. Between aspiration and reservation level the values of this function coincide with the membership function as well as have an ordering properties. In other segments it is used only for ordering alternatives. In order to properly handle – within the framework of the component achievement function – the criteria's values worse than a reservation level, and better than an aspiration level, it is necessary to allow for values of a component achievement function that are negative or greater than one; thus the (strict) MF can be understood as the projection of values of the component achievement function (or extended membership function) on the interval [0,1]. This function was used because practical applications show that quite often the user specifies non-attainable reservation levels and/or attainable aspiration levels. In such cases the optimization problem has nonunique solution.

The piece-wise linear component achievement functions (8) conform to the requirements for the extended valued membership functions formulated in [GrW94]. Note that the condition (11) corresponds well to the nature of the problem since one accepts small changes of u_i when a criterion value is better or close to an aspiration level. The speed of such change should increase along with moving towards a reservation level and should increase even faster between reservation and nadir points. Such features are consistent with the commonly known properties of the MF used in applications based on the fuzzy set approach.

Therefore there are many similarities between the ARBDS and the *Fuzzy Multi-objective Programming* approaches. The main difference is due to the specification and use of MF. The Fuzzy Multi-objective Programming method requires prior specification of aspiration and reservation levels which are used for the definitions of MF's. It is implicitly assumed that the criteria values for the all interesting solutions are between the corresponding aspiration and reservation levels (because the applied MF does not differentiate between solutions with values better than aspiration level and between those with values worse than reservation level). The user interactively specifies the reference membership levels for each MF, which can be interpreted as a degree of achievements of the aspiration for each criterion (scaled by the difference between aspiration and reservation).

The ARBDS method does not use the MF directly. It assumes that the user may change aspiration and reservation levels during the interaction upon the analysis of previously obtained solutions. The user specifies interactively the preferences in the space of the criteria values which seems to be more natural than a specification of preferences in the space of MF (in terms of degrees of achievements of MF values). A selection in the criteria space can however be interpreted in terms of Fuzzy Sets by a definition of a MF for a linguistic variable (e.g. *good solution*) for each criterion and an ex post interpretation to what degree a solution belongs to a set of *good* solutions. There is no need for restrictions[5] for the specification of aspiration and reservation levels in the criteria space. This is important for the analysis of large-scale complex problems for which specification of attainable reservation levels might be difficult.

2.3 Types and statuses of criteria

Both SAP and LP-Multi distinguish between the criterion type and status. The type of a criterion is defined during the preparatory stage (see Section 4.2) and can not be changed during the interaction. However, quite often the user wants to temporary treat a criterion in a different way. This can be achieved by changing the status of the criterion.

The `default` status of the criterion means that a criterion is treated according to its type originally defined by the user. The user may freely change the status of a criterion to one of: `stabilized, inactive, disregarded` and/or back to the `default` one (see Section 4.3.3 for details).

3 Structure and functions of the DSS shell

A Decision Support System has to be problem specific. However, the reuse of developed software is a rational way for implementation of new applications. Therefore, typically a model based DSS is composed of a number of mutually linked modules (cf [Mak94c] for a more detailed discussion). This Section provides an outline of the structure of the DSS shell, which has been developed for analysis of LP and MIP models using the ARBDS. A typical configuration of a DSS is illustrated in Figure 1 by the application developed for Regional Water Quality Management for the Nitra River Basin (cf [MSW95a] for details). One should note, that the illustration of the application of the DSS shell to the Land Use for Sustainable Agricultural Development Planning (cf [AFM95]) would differ only by the type of optimization problem (LP instead of MIP) and the solver used.

The DSS shell is composed of a number of modular and portable software tools that are characterized below with brief descriptions of their functions:

- A Graphical User Interface (GUI), which handles all the interaction with the user. GUI hides the differences between modules of the DSS from the user by providing a uniform way of interaction with all the components of the DSS.
- A problem-specific model generator (documented in [MSW95a]) for generating the core model which relates waste water emissions, selections of treatment technologies, and the resulting ambient water quality. It is important to stress that the core model includes

[5]For practical reasons the SAP constrains the choice between values of utopia and approximation of nadir. This is however not a real restriction since one should not expect to obtain the criteria values outside of this range.

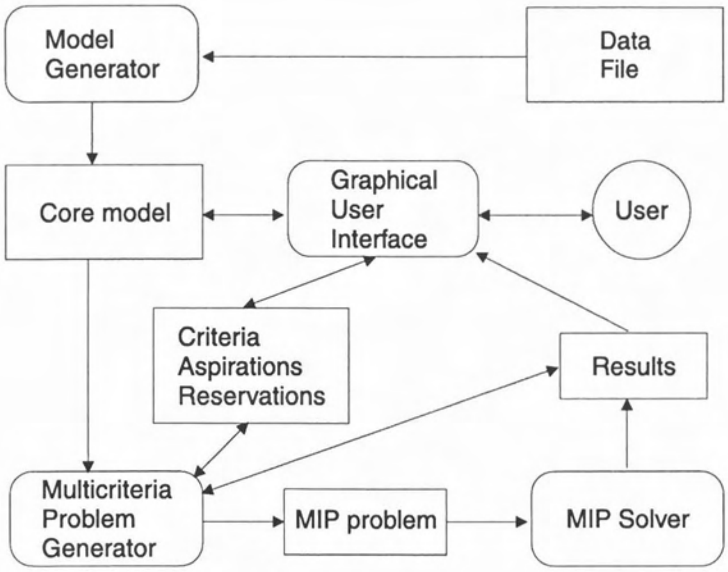

Figure 1: The structure of a Decision Support System for the water quality management in the Nitra River Basin.

only physical and logical relations, and not the preferential structure of the DM. A more detailed discussion on core model specification is provided in [Mak94c].

- The SAP described in this paper which supports specification of user preferences both in terms of aspiration/reservation levels and in terms of fuzzy sets. SAP also provides the user with other means of control over the problem analysis by allowing to change the criteria status, selection of displayed solutions, etc. In terms used in Figure 1 the SAP is used for the definition of `Aspirations, Reservations` and for changing the status of `Criteria`. However, the SAP provides more functions than can be outlined in Figure 1.
- The LP-Multi (see [Mak94c] for details), a modular tool for handling multiple criteria problems using the methodology outlined in Section 2. The resulting Mixed Integer Programming (MIP) problem is based on the core model and the aspiration and reservation levels which represent current preferential structure of a DM.
- A modular solver for mixed integer programming problems MOMIP described in [OgZ94]. The solver should be robust because in a typical application it is hidden from the user. Therefore a solver used in a DSS must not require interaction with the user.
- A data interchange tool LP-DIT described in [Mak94b]. This tool provides an easy and efficient way for the definition and modification of LP and MIP problems, as well as the interchange of data between a problem generator, a solver, and software modules which serve for problem modification and solution analysis. LP-DIT is used for the definitions of the core model and the MIP problems (the latter defined for each multiple-criterion problem), as well as for the optimization results.

The portability of the developed tools is achieved by using C++ programming language and a commercial tool for development of the portable Graphical User Interface (GUI), namely zApp library [Inm95] and the zApp Interface Pack [Inm93]. Modular structure and portability allow for the reuse of most of the components needed for a DSS applied to other problems. It also facilitates experiments with different solvers and with modules providing problem specific interaction with the user. Note, that a new application only requires the development of a model generator and optionally, a problem specific module for a more detailed analysis of results.

SAP is a module that can be used as a part of a model based DSS using a multicriteria optimization with aspiration and reservation levels. SAP plays a central role in the interaction with the user by providing all the functions necessary for interactive analysis of solutions and for specification of a new multicriteria optimization problem, namely:

- Specification of an aspiration and a reservation level.
- Optional specification of a piece-wise linear membership function for criteria values between aspiration and reservation levels.
- Changing the status of a criterion by stabilizing a criterion (minimizing a deviation of the criterion value from a given target value) or temporary disregarding a criterion.
- Supporting analysis of previously computed solutions by handling of the solution's history.

SAP has been designed and implemented with an inexperienced user in mind. Therefore the use of SAP by a person familiar with a window system is easy and does not require any substantial amount of training.

SAP has also been applied as a part of other Decision Support Systems to case studies that require analysis of nonlinear models (see Section 6) without changing a single line of code. This illustrates well the power of modular tools developed in C++.

4 User guide to SAP

This Section contains the following four groups of information assembled in the corresponding subsections:

- The general implementation assumptions that are of interest to any user of SAP,
- The preparatory stage of the problem analysis,
- Information about the interaction with SAP, which is useful for the user, who is not interested in technical details of the implementation of SAP.

4.1 Implementation assumptions

The design and implementation of SAP result from the requirements for its functions summarized in Section 3 and the underlying methodology described in Section 2. Additionally, it has been assumed that SAP has to be a modular tool, easily portable for different operating systems and usable for different applications.

The portability has been achieved by using C++ programming language and a portable commercial tool for Graphical User Interface (GUI). The *zApp Portable Application Framework* [Inm95] and *zApp Interface Pack* [Inm93] are used for GUI. These libraries are available for a number of operating systems therefore, the SAP can be easily ported at a moderate cost of purchasing a *zApp* version for another platform. Currently, the SAP has been developed for Solaris 2.4 and it will be ported soon to the MS-Windows.

SAP has been implemented as a C++ class conforming to the draft ANSI C++ specification. However templates and exception handling have not been used because these two features of C++ are still not supported by many compilers in a robust way. Therefore SAP can be used with applications written in ANSI C, C++ or Fortran. Use of C++ is strongly preferred, but the authors have enough positive experiences with linking C++ and Fortran code, therefore using SAP with an application written in Fortran should also be possible.

4.2 Preparatory stage

In order to allow for a Multi-Criteria Model Analysis (MCMA) SAP has to be used within an application, typically composed of a number of modules. In this paper we illustrate the use of SAP using as an example the DSS shell outlined in Section 3. The corresponding program is called mcma (from Multiple Criteria Model Analysis). Its main menu is presented in Figure 2 and a tutorial for its use is given in Appendix A.

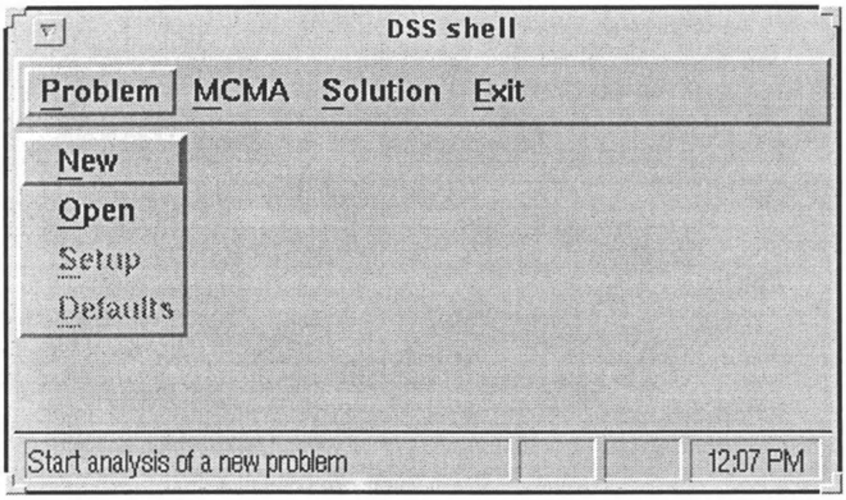

Figure 2: Main menu of the `mcma` application.

As already discussed in Section 2 the first stage of the problem analysis is building a corresponding *core model*. In order to be properly used in MCMA, the core model has to conform to a number of requirements, which are discussed in more detail in [Mak94c]. Assuming that the core model is available in the form of LP-DIT binary files (models available in the form of MPS format file can be converted by the mps2dit utility available with the LP-DIT library [Mak94b]) one can start execution of mcma. The main menu of this application is reproduced in Figure 2. At this point only two menu items have active[6] submenu items: `Problem` and `Exit`.

The `Problem` menu contains the following submenu items, selection of which will result in the actions described below:

`New` – starts analysis of a new (or modified) core model. The problem is selected by a choice (from the presented list of files in the current directory having the extension `cor`) of a file containing a core model definition in the LP-DIT format. The root of the file name defines the problem `id` that is used for the definition of file names generated during the problem analysis (see Section 4.2.3 for details).

`Open` – opens a problem for a continuation of the analysis. The problem is selected by a choice (from the presented list of files in the current directory having the extension `mc`) of a file. The information about a previously made analysis is stored in the file `id.mc` (where `id` has been defined by the core model selection made for the corresponding `New` problem). Therefore a selection of such a file results in continuation of the analysis done at an earlier time. In order to continue the analysis of the problem, the corresponding

[6] Non-active submenu items (i.e. items corresponding to actions that are not available at a certain point of the interaction) are grayed. Clicking on a grayed item is ignored.

core model must not be modified; hence the `id.cor` file should neither be modified nor removed from the current directory.

`Setup` – advanced users can use this option for redefining options for a solver used with the application.

`Defaults` – resets the options for a solver to the default values set for the application by its developer.

For the analysis of a new (or modified) problem the core model definition has to be provided in the LP-DIT format in the file `id.cor`, where `id` is the implied problem name. The next step (forced by LP-Multi) is the definition of criteria which is done interactively (see Section 4.2.1). After criteria are defined the LP-Multi starts a series of optimizations in order to compute the Utopia point and an approximation of the Nadir point. Finally a compromise Pareto-optimal solution is computed. The preparatory stage for a new problem requires $2n + 1$ optimization runs (where n is a number of criteria).

After either a set of criteria for a new problem is defined and the compromise solution is found, or a previously analyzed problem has been open, the user can start SAP by selecting the `MCMA` main menu item. Note that the window of the main menu is used for displaying messages that inform about the execution of the program and do not require action nor confirmation of the user. The messages that require a reaction from the user are displayed in pop-up windows.

4.2.1 Definition of criteria

Criteria are defined interactively, if a new problem is selected. A criterion is defined by the variable of a core model and the type. In order to facilitate selection of variables for medium-sized and large models, the definition of criteria is split into two stages:

- First, the user interactively selects from a list of variables those which will be used for criteria. The dialog for selecting variables has the help button which provides the necessary information. In order to avoid scrolling a long list of all variables of the core model, one can define a mask (first characters of names of variables) for variables that are displayed for selection. Several masks can be defined sequentially for providing a list composed of names that start with different characters. For small models the user can define an empty mask, which will result in including all the variables into the selection list.
- Second, the user selects for each variable a criterion name and its type. The criterion name is limited to 6 characters and in order to conform to the requirements of some solvers must be composed of printable characters which do not include blank spaces.

The current implementation of LP-Multi and SAP allows for three types of criteria: `minimized, maximized and goal`. The meaning of first two types is obvious but the `goal` type requires an explanation.

The `goal` type of a criterion should be used only if the meaning of a criterion is such that a criterion should have a given goal (target) value and should neither be minimized nor maximized. Note that the status of a minimized or maximized criterion can be switched during the MCMA to a stabilized criterion, which has exactly the same meaning as a `goal` type of a criterion. Afterwards, the status of a stabilized criterion can be changed back to its original type, except for the criteria of `goal` type that always have the status `stabilized`. However, a criterion type, once defined in the preparatory stage of MCMA, can not be changed. The distinction between the `goal` type and the `stabilized` status is forced by the need of assuring the consistency of MCMA.

For a `goal` type criterion the user has to interactively specify an initial target value. The Utopia point component for such criterion is the minimal attainable deviation from this value. This value can be changed during MCMA, therefore for such a criterion (contrary to other types of criteria) the Utopia value can also be changed during MCMA. This might be misleading for some users. The definition of a `goal` type criterion is rarely needed. However, there are situations in which this type is useful (for example for analysis of soft equality constraints) and has a clear interpretation.

4.2.2 Information stored for further analysis

Quite often a more detailed (possibly problem specific) analysis of some solutions is desired in a non-interactive way. For that purpose the following information is available:

- The user can save complete information about the current solution by selecting the `Store` item from the `Solution` menu item of the main window (see Figure 2). The file contains the MPS-like text formatted solution (the format is solver dependent) augmented by the information about the criteria values, and aspiration and reservation levels. Full text solutions are not stored automatically (due to the size of the full solution text files) and are overwritten by the solution of the next optimization problem.
- For each solution computed during the problem analysis one line of information is stored in the `summary` file (see Section 4.2.3). This line contains the following fields (separated by the | character): sequence number of the solution, triples with criterion value, aspiration and reservation levels for each criterion, and the name of the file with the complete solution generated by a solver.

4.2.3 Files created during the analysis

There are a number of files that are generated during the analysis. All file names have the problem `id` (see Section 4.2) as the root of name, and the extension of the name identifies the files with the following contents:

`cor` – core model in LP-DIT format (this is the only file that has to be provided by the user)[7].

`mc` – binary file with information used by LP-Multi.

`xxx` – text files, each containing a full solution stored by the user during the interaction; `xxx` is a three digit number corresponding to the sequence number of the optimization problem.

`sum` – a text summary file containing one line information about each solution.

`ft` – history file maintained by SAP.

`lp` – LP/MIP optimization problem in LP-DIT format.

`sol` – solution of the last optimization problem in LP-DIT format.

`txt` – solution of the last optimization problem in text format.

`log` – log file produced by a solver.

`lg2` – second (if any) log file produced by a solver.

4.3 Interaction with SAP

The main SAP window (see Figure 3) can be created after a preparatory stage described in Section 4.2 is completed. The window is created by the selection of the `SAP` item from the

[7]For models available in the MPS format one can use the `mps2dit` utility provided with the LP-DIT library.

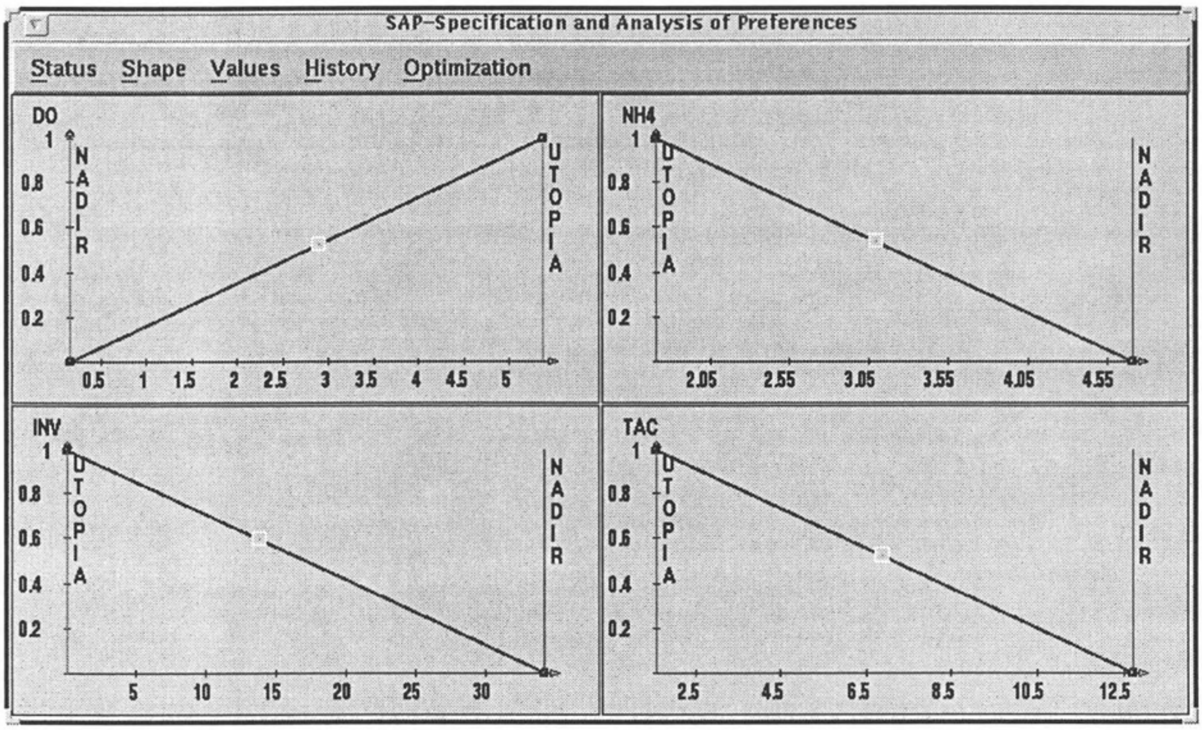

Figure 3: Main window of SAP.

MCMA menu available from the main menu (see Figure 2). There are a number of functions that can be selected from the menu of the SAP main window. The most commonly used (and therefore a default) function is described in Section 4.3.1. Other groups of actions that can be activated by the selection of a SAP menu item are described in the following subsections.

4.3.1 Specification of aspiration and reservation levels

The default function of SAP is a selection of aspiration and reservation levels for each active criterion (Section 4.3.3 describes changing of a criterion status). For each criterion the last specified component achievement function $u_i(q_i)$ is plotted. The aspiration and reservation levels can be set by using either the mouse or a keyboard. Clicking the mouse results in moving the nearest (either aspiration or reservation) point to the point which the mouse is currently pointing. This is the easiest and fastest way of setting those values. The exact values for aspiration and reservation levels can be displayed and updated from a pop-up window, which can be created by selection of the `Values` item from the SAP menu.

Aspiration and reservation levels must not be too close (cf. [Mak94c] for more detailed information). Should the specified values for those levels for a criterion be close, then either they should be modified or the criterion status has to be changed for the stabilized one. Therefore, if such a case is detected, the user is asked to either change the aspiration and/or reservation levels or the status of the criterion.

4.3.2 Specification of preferences in terms of fuzzy sets

The user can specify his/her preferences for the criteria values between aspiration and reservation levels by specification of a piece-wise linear function (see Section 2.2) which can also be interpreted as an extended valued membership function. In order to create or delete additional points a corresponding item from the `Shape` menu should be selected (see Figure 4 for the illustration). The points are added or deleted (depending on the

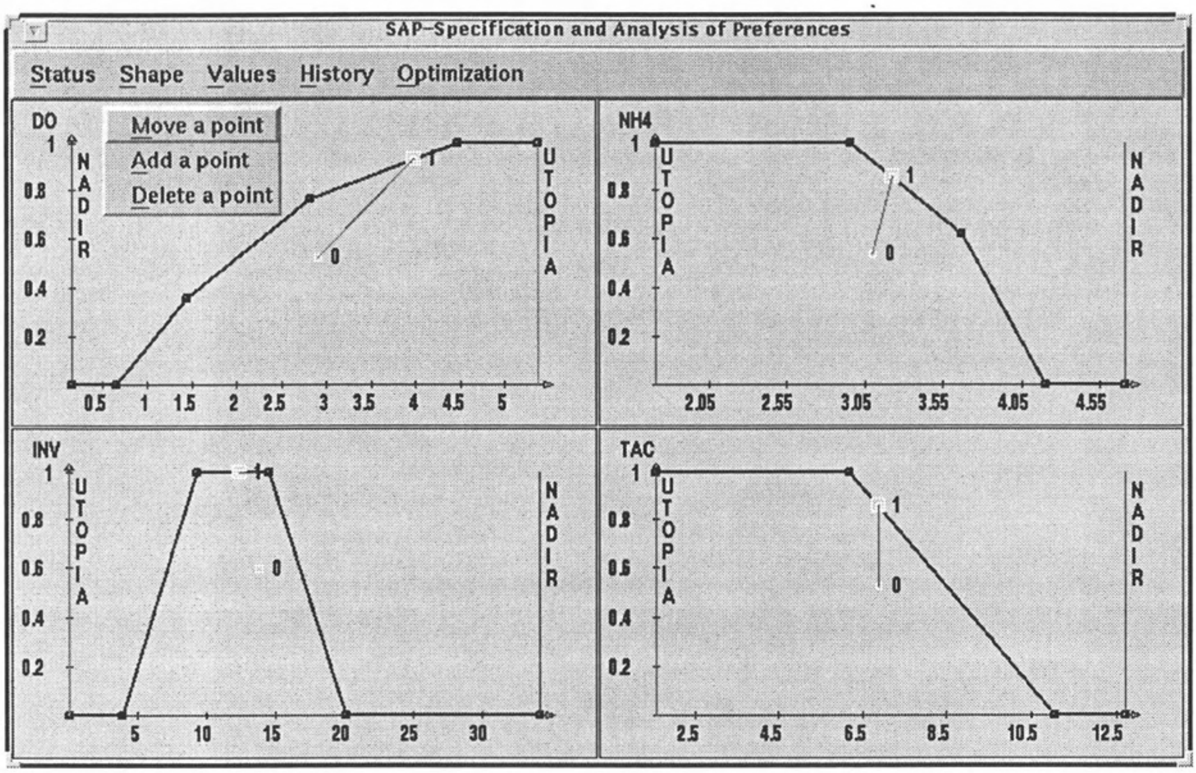

Figure 4: Shape menu and PWL function.

selection made from the `Shape` menu) until another selection is made from the `Shape` menu.

Note that the conditions (12,13) are forced by SAP. However, the user may specify a function that does not fulfill condition (11). In such a case SAP removes the points that cause non-concavity of the function and informs the user about the modifications. The user may either accept the changes made by SAP or he/she may make further modifications.

4.3.3 Status of criteria

Figure 5 illustrates the dialog (activated by the `Status` item from the SAP menu) for changing the status of each criterion. The user can change (by pressing a corresponding radio button) the status of a criterion to one of the following:

`min/max/goal` – a criterion has its default status (a status originally defined by the user as described in Section 4.2.1).

`stabilized` – a minimized or maximized criterion is converted as a stabilized criterion (see Section 4.3.4). This selection is suppressed for a `goal` type of a criterion.

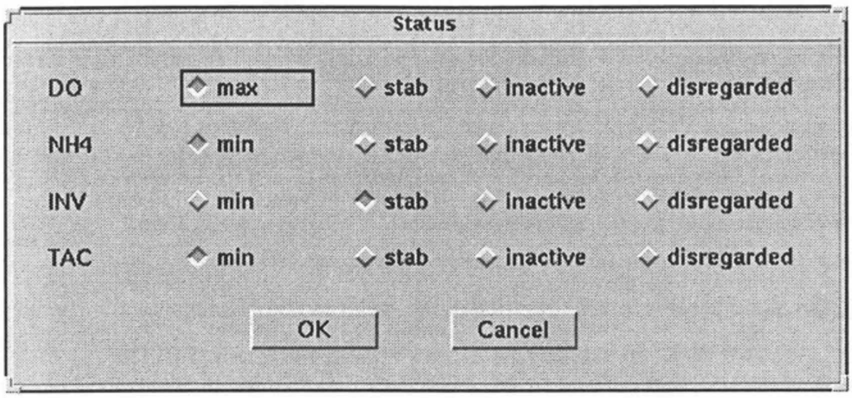

Figure 5: Dialog for changing criteria status.

`inactive` – a criterion is temporary disregarded and its component achievement function is not defined. However, a criterion enters the scalarizing function (3), because the s_i coefficient for such criterion is defined by (4).

`disregarded` – a criterion is completely dropped from entering the scalarizing function.

4.3.4 Goal type and stabilized criteria

A `goal` type criterion (see Section 4.2.1) and a criterion whose status has been changed to `stabilized` criterion are treated in the same way, with only one exception: for a `goal` type criterion the user can specify any target value, whereas for a `stabilized` criterion a target value must be between the corresponding Utopia and Nadir components. Hence we will also use the term `stabilized` for a `goal` type criteria in this subsection. For the sake of brevity we will ignore the index i of a criterion q_i and of the corresponding quantities (target value, aspiration, reservation levels) in this subsection.

For a stabilized criterion one has to select a target (desired) value $\tilde{q}$ and to specify two pairs of aspiration and reservation levels, which are interpreted as *still accepted* and *no longer accepted* values of the criterion. One pair is defined for a surplus (over the target value $\tilde{q}$) and the second pair is for a deficit (values lower than $\tilde{q}$). The pairs of aspiration and reservation levels are denoted by $(\bar{q}^+, \underline{q}^+)$ and $(\bar{q}^-, \underline{q}^-)$, respectively.

For many criteria the deviations from the target value in both directions (surplus and deficit) have similar meaning, therefore it is useful to distinguish cases in which a target value is equal to a mean of aspiration levels. This type of a stabilized criterion is called the `symmetric` type and the following condition is forced by the way in which the interaction is implemented:

$$\tilde{q} = (\bar{q}^+ + \underline{q}^+)/2. = (\bar{q}^- + \underline{q}^-)/2. \qquad (14)$$

The `symmetric` type is the default type for a stabilized criterion. It can be changed from the `Shape` menu to the `general` type for a criterion in which the user does not accept the condition (14) and wants to specify aspiration and reservation levels independently for surplus and deficit. The `general` type of a stabilized criterion provides much more flexibility at the expense of a more time consuming interaction. Note that one can optionally define a piece-wise linear MF for both (`symmetric` and `general`) types of a stabilized criterion.

Processing stabilized criteria is implemented by the generation of additional criteria. This is done by LP-Multi in a way that is transparent to the user. Details of the processing are documented in [Mak94c].

4.3.5 History

View Solutions

	sol. nr.	DO	NH4	INV	TAC
A	0	2.885923	3.098945	13.8	6.890943
B	1	4.252989	3.257073	8.2	7.103169
C	2	4.232047	3.520948	4.7	6.19206
D	3	3.231534	3.230792	13.5	6.495705
E	4	3.982716	3.089756	12.5	7.068245
F	5	4.942418	2.392354	22	9.774112

Figure 6: The history in the form of a spreadsheet.

SAP keeps record of all the Pareto-optimal points and the corresponding aspiration and reservation levels. The history of all solutions can be examined in the form of a spreadsheet (see Figure 6 for the illustration) that can be displayed by the `History` item from the SAP menu. The records are arranged in the following way:

- First, `M` solutions that are displayed in the main SAP windows. The number `M` is set by SAP to be equal to 10, but this value can be changed by the user. Each solution has a sequence number that is set by SAP and a label. The sequence numbers can not be changed but the user can change the labels of the displayed solutions. Both the labels and the number of displayed solutions `M` can be changed by the dialog activated by the `Setup` item from the `History` menu of SAP.
- Second, all the solutions that are currently not displayed in the SAP main window, sorted by their sequence numbers.

Solutions can be added to or removed from the set of displayed solutions by clicking on the corresponding sequence number. The last solution is automatically added to the displayed solutions, as long as there are empty slots in the set of displayed solutions. The user is asked to rearrange the set of displayed solutions, once this set is full.

The last (rightmost) field in the solution spreadsheet provides space for the user comment or notes related to a corresponding solution.

The selected `M` solutions are displayed in the main SAP window (see Figure 7 for the illustration). Previously obtained solutions are marked by small squares with the labels (the default labels shown in Figure 7 can be changed as described above). The last obtained solution is marked by a larger square and is connected with the previous solution by a thin line.

4.3.6 Optimization

The user should select this menu item, when he/she has finished the specification of a new multiple-criteria optimization problem. Once the `Run` item from the `Optimization` menu is selected the following actions are performed:

- The last solution is added to the History records.
- If there is a space in the Displayed History Set, then the last solution is added to this set. Otherwise the user is asked to update this set.

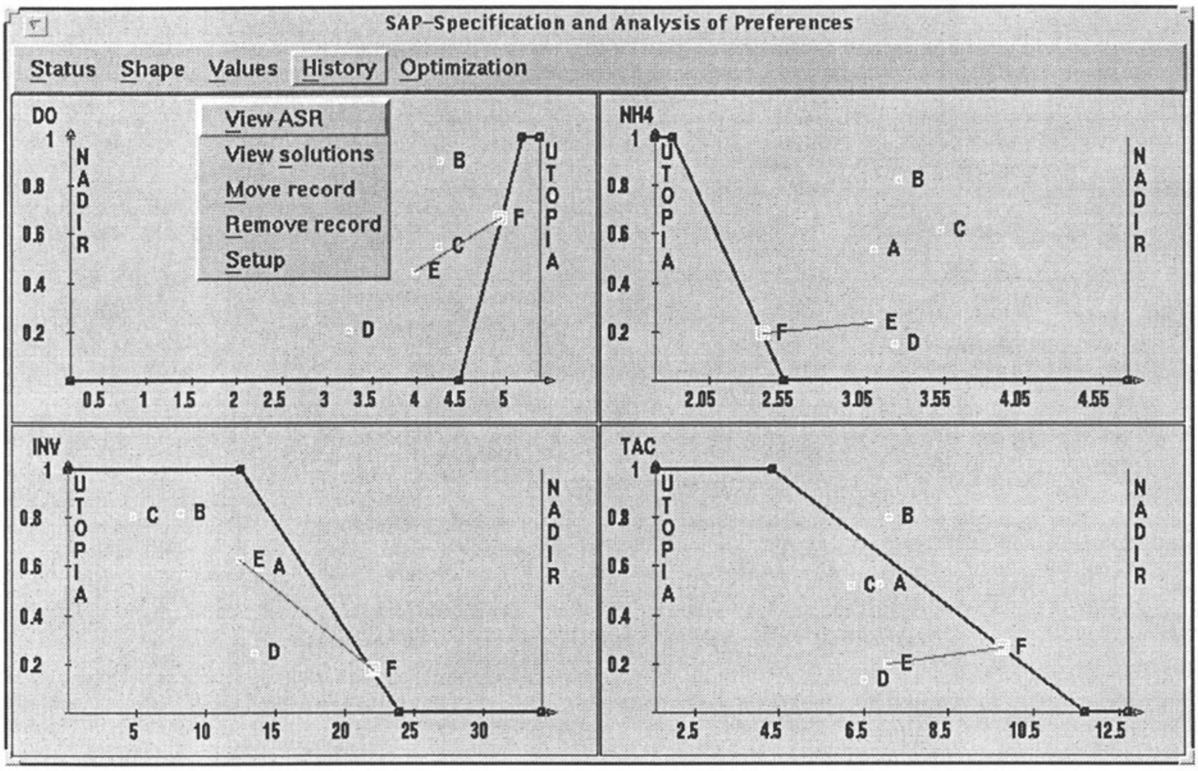

Figure 7: The history in the form of plots.

- The control of the program is turned over to LP-Multi. The optimization problem is generated and solved. The last solution is processed and converted to the SAP data structures.
- The control is transferred back to the user, who can either make a new iteration of MCMA or break the analysis.

In the current implementation of mcma one of the following two solvers can be used:

- HOPDM (see [GoM95]), interior point based LP solver, especially efficient for medium and large scale problems.
- MOMIP (see [OgZ94]), modular optimizer for Mixed Integer Programming.

The default selection of the solver is done by mcma. HOPDM is chosen for LP optimization problems and MOMIP for MIP problems. The set of solver parameters is selected in order to allow for efficient solving of a broad class of respective types of optimization problems. The user is advised to contact the authors should the solution time became unacceptable. A problem specific tuning of solver parameters may substantially improve the performance of a solver.

5 Availability of software and documentation

SAP together with LP-Multi can be easily used for analysis of LP and MIP models at IIASA and at Institute of Control and Computing Engineering, Warsaw University of Technology, provided that a corresponding *core model* is available in the LP-DIT format [Mak94b] or in the MPS format. A beta version of the SAP is also available by anonymous ftp (upon e-mail request) in a form of executable shell DSS, free of charge for non-commercial research and educational purposes. The DSS shell can be used in a way illustrated by a tutorial session (see Appendix A) to the analysis of a core model (available in either LP-DIT or MPS format). The distributable set also contains two solvers (HOPDM and MOMIP) and two core models (corresponding to the Nitra and Land Use case studies (see Section 6 for details). At the time of writing this text only the Solaris 2.4 version of SAP is available. A versions for MS-Windows will be available in near the future.

This Working Paper serves as a documentation of SAP and it is primarily aimed at users of ready applications that include SAP. Updated versions of this Paper will be made available, if the need arises. All Working Papers published by the Methodology of Decision Analysis Project are available from the Publication Department of IIASA. Most of them (including all papers written by the authors of this WP) are available via the WWW of IIASA:
`http://www.iiasa.ac.at`
The Welcome Page of the IIASA WWW provides an easy access to the IIASA Publications, which can be examined in various ways (by author's name, project, date, etc). Postscript files can be obtained free of charge via WWW. Hard copies can be ordered from the Publication Department of IIASA (orders can be placed also via WWW).

Developers who want to include SAP into an application should contact the authors for additional information. However, in order to generate a problem specific application which uses SAP one has to have the *zApp Library* [Inm95] and the *zApp Interface Pack* [Inm93].

6 Conclusion

Until now, the SAP has been implemented within the following applications:

- A DSS developed for the Regional Water Quality Management Problem, case study of the Nitra River Basin (Slovakia) documented in [MSW95a, MSW95b]. This application is a result of cooperation of MDA and WAT Projects.
- Multiple Criteria Analysis in Optimizing Land Use for Sustainable Agricultural Development Planning (cf [AFM95]). This application is a result of cooperation of LUC and MDA Projects with the FAO (Food and Agriculture Organization of the United Nations).
- A number of engineering applications in mechanics, automatic control and ship navigation (summarized in [WiG96]).

Other applications are planned in the near future.

The following extensions of SAP are planned (the sequence corresponds to the current priorities set by the authors):

- Graphical comparison of selected solutions.
- Interactive analysis of full solution.
- Interface to the interactive definition of soft constraints.
- Printing of the contents of the SAP window.
- Analysis of history using an extension of the methodology described and applied for BIPLOT (cf [LeG91].

The authors would appreciate comments and suggestions regarding functionality and robustness of SAP. Please do not hesitate to contact one of the authors (preferably by e-mail: `granat@ia.pw.edu.pl` or `marek@iiasa.ac.at`) if more information is desired.

References

[AFM95] J. Antoine, G. Fischer and M. Makowski, *Multiple criteria analysis in optimizing land use for sustainable agricultural development planning*, Working Paper WP-95-xx, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1995. (forthcoming).

[ChC67] A. Charnes and W. Cooper, *Management Models and Industrial Applications of Linear Programming*, J. Wiley & Sons, New York, London, 1967.

[EKO90] H. Eschenauer, J. Koski and A. Osyczka, eds., *Multicriteria Design Optimization: Procedures and Optimization*, Springer Verlag, Berlin, Heidelberg, New York, 1990.

[GaS94a] L. Gardiner and R. Steuer, *Unified interactive multiple objective programming*, European Journal of Operational Research **74** (1994) 391–406.

[GaS94b] ______, *Unified interactive multiple objective programming: An open architecture for accommodating new procedures*, Journal of Operational Research Society **45**, no. 12 (1994) 1456–1466.

[GoM95] J. Gondzio and M. Makowski, *HOPDM, modular solver for lp problems; User's guide to version 2.12*, Working Paper WP-95-50, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1995.

[GrW94] J. Granat and A. P. Wierzbicki, *Interactive specification of DSS user preferences in terms of fuzzy sets*, Working Paper WP-94-29, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994.

[Inm93] Inmark Development, Mountain View, CA, USA, *zApp Interface Pack, Programmer's Guide & Reference*, 1993.

[Inm95] Inmark Development, Mountain View, CA, USA, *zApp, The Portable C++ Application Framework, Programmer's Guide*, 1995.

[IsS87] H. Isermann and R. E. Steuer, *Computational experience concerning payoff tables and minimum criterion values over the efficient set*, European J. Oper. Res. **33** (1987) 91–97.

[KLW91] P. Korhonen, A. Lewandowski and J. Wallenius, eds., *Multiple Criteria Decision Support*, Lecture Notes in Economics and Mathematical Systems, vol. 356, Springer Verlag, Berlin, New York, 1991.

[KoW89] P. Korhonen and J. Wallenius, *Observations regarding choice behaviour in interactive multiple criteria decision-making environments: An experimental investigation*, in Methodology and Software for Interactive Decision Support, A. Lewandowski and I. Stanchev, eds., Lecture Notes in Economics and Mathematical Systems, vol. 337, Springer Verlag, Berlin, New York, 1989, pp. 163–170.

[LeG91] A. Lewandowski and J. Granat, *Dynamic* BIPLOT *as the interaction interface for aspiration based decision support systems*, in Multiple Criteria Decision Support, P. Korhonen, A. Lewandowski and J. Wallenius, eds., Lecture Notes in Economics and Mathematical Systems, vol. 356, Springer Verlag, Berlin, New York, 1991.

[LeW89a] A. Lewandowski and A. Wierzbicki, eds., *Aspiration Based Decision Support Systems: Theory, Software and Applications*, Lecture Notes in Economics and Mathematical Systems, vol. 331, Springer Verlag, Berlin, New York, 1989.

[LeW89b] A. Lewandowski and A. Wierzbicki, *Decision support systems using reference point optimization*, in Aspiration Based Decision Support Systems: Theory, Software and Applications, A. Lewandowski and A. Wierzbicki, eds., Lecture Notes in Economics and Mathematical Systems, vol. 331, Springer Verlag, Berlin, New York, 1989.

[Mak94a] M. Makowski, *Design and implementation of model-based decision support systems*, Working Paper WP-94-86, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994.

[Mak94b] ———, *LP-DIT, Data Interchange Tool for Linear Programming Problems, (version 1.20)*, Working Paper WP-94-36, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994.

[Mak94c] ———, *Methodology and a modular tool for multiple criteria analysis of LP models*, Working Paper WP-94-102, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994.

[MSW95a] M. Makowski, L. Somlyódy and D. Watkins, *Multiple criteria analysis for regional water quality management: the Nitra River case*, Working Paper WP-95-22, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1995.

[MSW95b] ———, *Multiple criteria analysis for water quality management in the Nitra basin*, Water Resources Bulletin (1995). Submitted in May 1995.

[Nak94] H. Nakayama, *Aspiration level approach to interactive multi-objective programming and its applications*, Working Paper WP-94-112, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994.

[OgL92] W. Ogryczak and S. Lahoda, *Aspiration/reservation-based decision support — a step beyond goal programming*, Journal of Multi-Criteria Decision Analysis **1**, no. 2 (1992) 101–117.

[OgZ94] W. Ogryczak and K. Zorychta, *Modular optimizer for mixed integer programming,* MOMIP *version 2.1*, Working Paper WP-94-35, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1994.

[Sak93] M. Sakawa, *Fuzzy Sets and Interactive Multiobjective Optimization*, Plenum Press, New York, London, 1993.

[SeS88] F. Seo and M. Sakawa, *Multiple Criteria Decision Analysis in Regional Planning: Concepts, Methods and Applications*, D. Reidel Publishing Company, Dordrecht, 1988.

[Ste92] T. Stewart, *A critical survey on the status of multiple criteria decision making theory and practice*, OMEGA, International Journal of Management Science **20**, no. 5/6 (1992) 569–586.

[WeW93] J. Wessels and A. Wierzbicki, eds., *User-Oriented Methodology and Techniques of Decision Analysis and Support*, Lecture Notes in Economics and Mathematical Systems, vol. 397, Springer Verlag, Berlin, New York, 1993.

[Wie77] A. Wierzbicki, *Basic properties of scalarizing functionals for multiobjective optimization*, Mathematische Operationsforschung und Statistik, s. Optimization **8** (1977) 55–60.

[Wie80] ———, *The use of reference objectives in multiobjective optimization*, in Multiple Criteria Decision Making, Theory and Applications, G. Fandel and T. Gal, eds., Lecture Notes in Economics and Mathematical Systems, vol. 177, Springer Verlag, Berlin, New York, 1980, pp. 468–486.

[Wie82] ———, *A mathematical basis for satisficing decision making*, Mathematical Modelling **13** (1982) 5–29.

[Wie86] ———, *On the completeness and constructiveness of parametric characterizations to vector optimization problems*, OR Spektrum **8** (1986) 73–87.

[Wie92] ———, *Multi-objective modeling and simulation for decision support*, Working Paper WP-92-80, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1992.

[WiG96] A. Wierzbicki and J. Granat, *Multi-objective modeling for engineering applications in decision support*, in Proceedings of the Twelfth International Conference on Multiple Criteria Decision Making, Lecture Notes in Economics and Mathematical Systems, Springer Verlag, Berlin, New York, 1996. (accepted for publication).

[Zim78] H. Zimmermann, *Fuzzy programming and linear programming with several objective functions*, Fuzzy Sets and Systems **1** (1978) 45–55.

[Zim85] ———, *Fuzzy Set Theory – and Its Applications*, Kluwer, Boston, Dordrecht, Lancaster, 1985.

[Zim87] ———, *Fuzzy Sets, Decision Making, and Expert Systems*, Kluwer, Boston, Dordrecht, Lancaster, 1987.

A Tutorial for using the distributable DSS shell

This appendix contains a tutorial session which illustrates the typical procedure of working with the DSS shell. The DSS shell is distributed with two core models:

- `nitra.cor` — Regional Water Quality Management for the Nitra River documented in [MSW95a].
- `aez.cor` — Agroecological Zone Model used for optimizing land use for sustainable agricultural development planning (see [AFM95]).

In order to became familiar with the basic functions of SAP we suggest the following steps:

1. At the shell prompt type: **mcma &** and press <enter>. The initial window will be displayed on the screen (Figure 8).

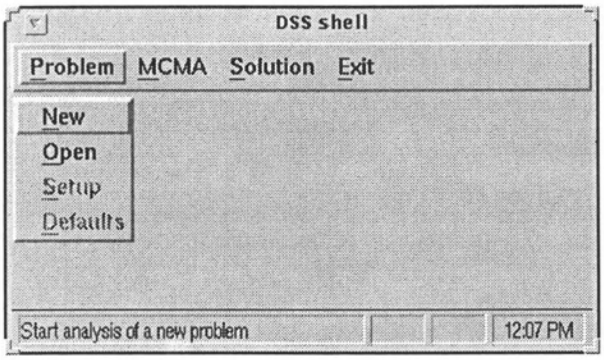

Figure 8: Main menu of the mcma application

2. Select the **New** option if you want to analyze the problem. The list of available core models will be displayed (Figure 9). Then select **nitra.cor** file and click the mouse on the **OK** button.

3. Next the dialog titled *Select a mask for a name* will be displayed (Figure 10). This gives an opportunity to preselect a group of variables with names that start which a given string (one or more characters). Typically a core model contains hundreds of variables out of which only few are used as criteria. Selection of an empty string in this dialog would result in providing all variables (about 800 for the `nitra.cor` core model).

4. In order to select only variables which can be used for environmental criteria type **cr** as a mask name and click the mouse on the **OK** button. The window titled *Variables defining criteria* will be displayed (Figure 11).

5. Double click on the name **cr_0** to copy this name from *Selected variables* list to *Criteria defined by:* list. Do the same for the name **cr_3**. You may also click on the **Help** button to read a short information about this dialog. The variables **cr_0** and **cr_3** represent two water quality constituents, namely DO (dissolved oxygen) and NH4 (ammonia), respectively.

6. In order to add to the list of variables those which define cost criteria click the mouse on the **Add** button. The dialog titled *Select a mask for a name* will be displayed

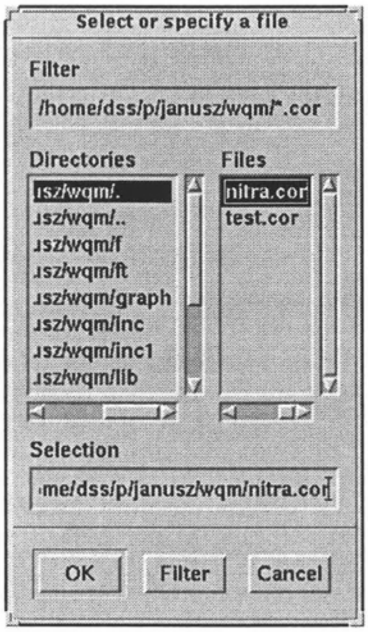

Figure 9: Select or specify a file dialog

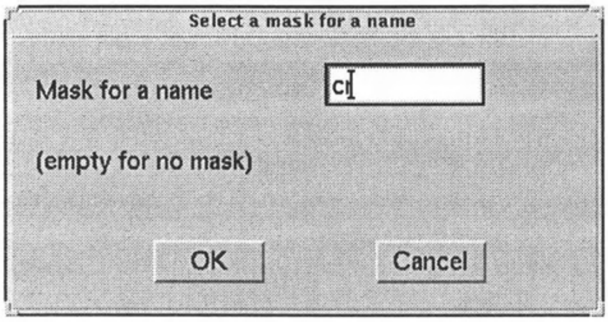

Figure 10: Select the mask **cr** for a name

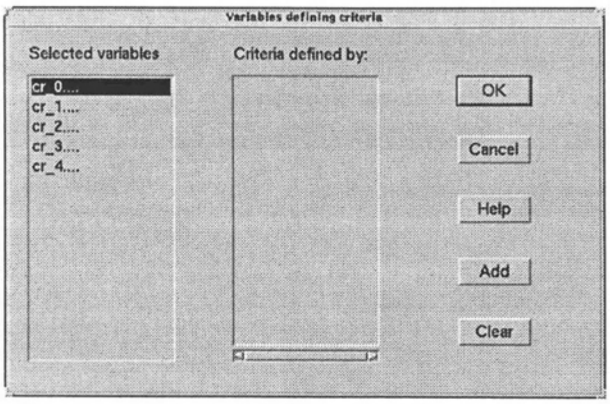

Figure 11: Variables defining criteria

(Figure 10) will be displayed again. Type **tot** as a mask name and click the mouse on the **OK** button. The dialog titled *Variables defining criteria* will be displayed again (Figure 12) with three more variables in the left window.

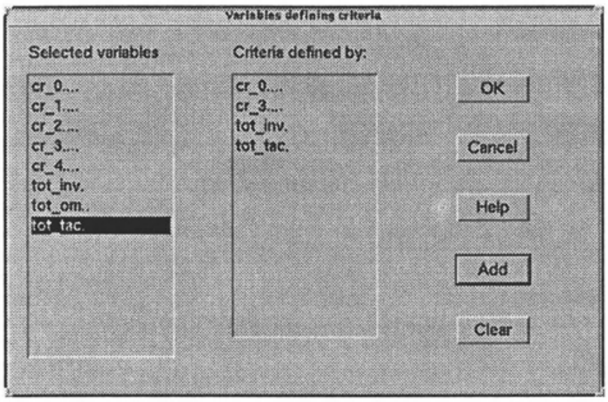

Figure 12: Variables defining criteria and selected criteria

7. Double click on the name **tot_inv**. Then the name **tot_inv** will be copied from *Selected variables* list to *Criteria defined by:* list. Do the same for the name **tot_tac** (see Figure 12). The variables **tot_inv** and **tot_tac** represents cost variables the total investment cost and the total annual cost respectively. Click the mouse on the **OK** button.

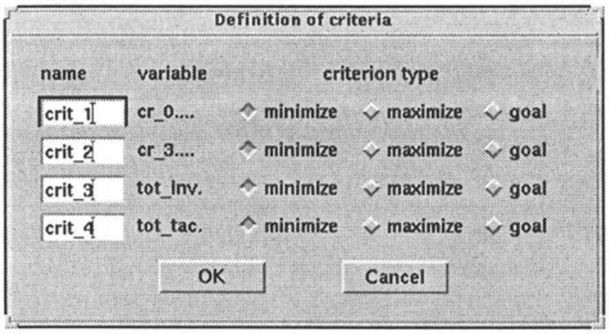

Figure 13: Definition of criteria dialog with the default criteria names and types.

8. Next step is to (optionally) define more meaningful names for criteria and to select for each criterion its type. This is done by the dialog titled *Definition of criteria* (Figure 13) which is displayed after the selection of variables defining the criteria. The names of criteria are predefined as `crit_?` (where ? is replaced by a digit). We suggest to use the following names: **DO** for **cr_0** criterion, **NH4** for **cr_3**, **INV** for **tot_inv**, and **TAC** for **tot_tac**. Note that (due to the MPS format restrictions adopted by many LP packages) the names of criteria are restricted to 6 characters (two more characters are needed by `LP-Multi` for creation of unique names of the parametric optimization problem). The `variable` column with the static strings contains the names of core model variables that define the corresponding criteria.

Specify the criterion type maximize for **DO** by clicking on the corresponding radio button (see Figure 14). Double check the criteria names and types and click the mouse on the **OK** button when you are done.

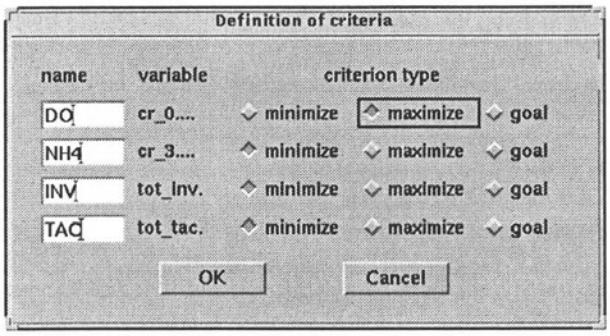

Figure 14: Definition of criteria dialog (after suggested modifications).

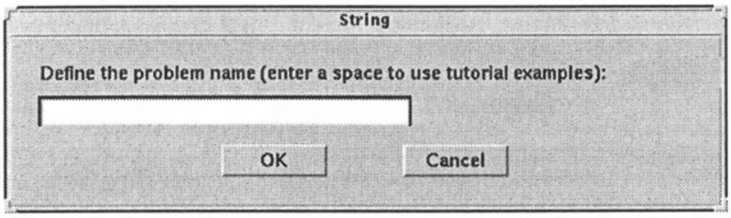

Figure 15: Problem name definition dialog.

9. The next dialog (Figure 15) will ask for the name of the problem. You can define your name or you can use the predefined name by pressing <space> and then <enter>. Completion of this dialog finishes the preparatory interaction with the user aimed at the definition of criteria. Then the **LP-multi** module starts a series of optimizations in order to compute Utopia point, approximation of nadir point, and a compromise Pareto-optimal solution. For the 4 selected criteria this will require solving 9 MIP problems each with about 800 variables and rows. You can follow the information about the generation of parametric single-criterion optimization problems and update values of the utopia point and approximation of the nadir point.

10. After the computation of the compromise solution has been completed (which you can also recognize by change of the default name of the main window of the DSS Shell to the name you selected for the problem) start SAP module by selecting the `MCMA` item from the SAP main menu. The main window of the SAP module will be displayed (Figure 16). For each of the defined criterion, the compromise solution marked by a rectangle will be presented. The aspirations levels are equal to utopia values and reservation levels are set to the nadir values. Now the interactive multicriteria model analysis may begin. One iteration consist of analysis of previous solutions, selection of new aspiration and reservation levels, and optimization.

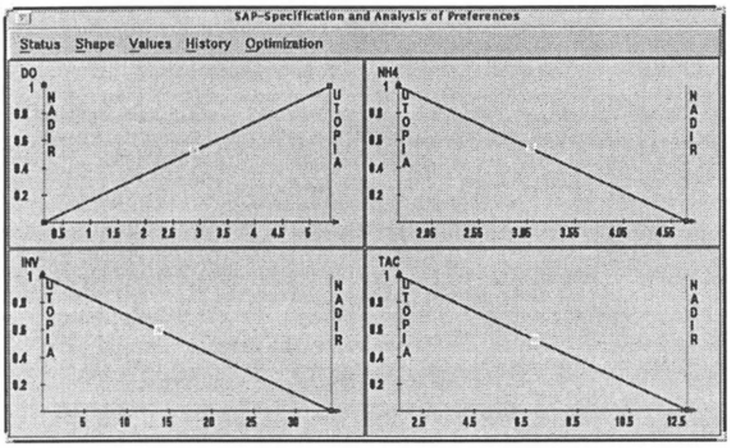

Figure 16: Interaction screen 1.

11. You may also change the the status of a criterion (e.g. of the criterion **INV**) by selecting the main menu the item **Status**. The dialog for changing the status is

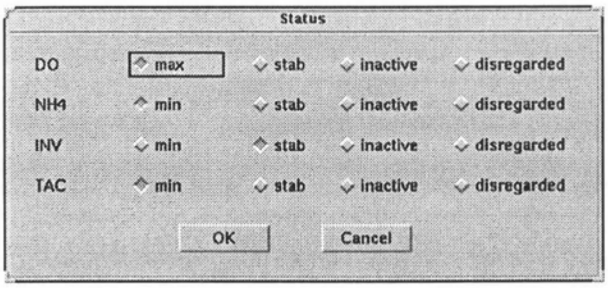

Figure 17: Dialog for changing criteria status.

presented in Figure 17. Set the new preferences of the user which are presented in Figure 18. The new aspiration and reservation levels for **DO**, **NH4**, and **TAC** can be set by clicking the mouse near the point which you want to specify. The final shape of the function for stabilized criterion can by obtained by clicking the mouse. Optionally, you can specify exact values of aspiration and reservation levels by using **Values** option.

12. Once you have decided the status of all criteria and the new set of aspiration and reservation levels for each criterion, select the `Optimization` item from the SAP main menu. This will start generation and solution of the new optimization problem. Figure 18 shows also the result of calculations. The rectangle marks the new solution and the thin line connects the last solution and the new one. The full information about the current solution can be saved by selecting the **Store** item from the **Solution** menu item of the main window.

13. In the subsequent iterations, the history of the process of interaction can by analyzed graphically or in the form of a spreadsheet. The small rectangles in Figure 19 represent the previously calculated solutions. The complete information about the previously optimization run is available in two forms of spreadsheets that can be

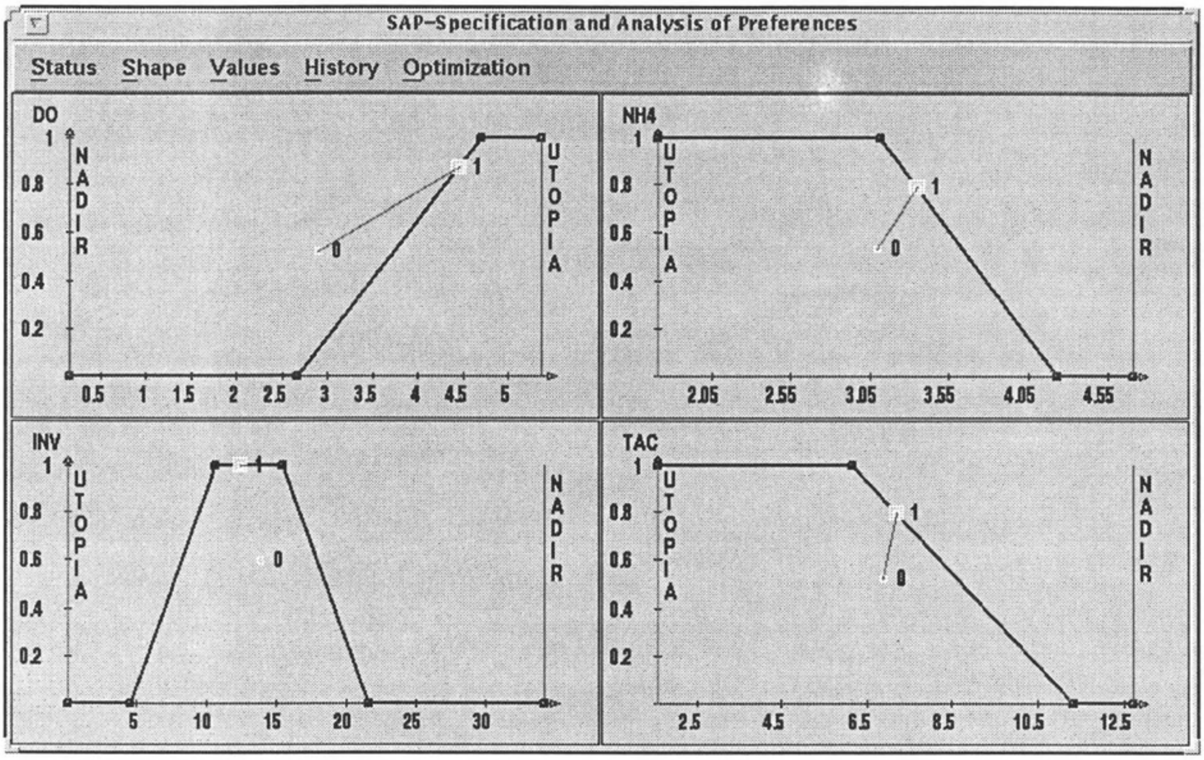

Figure 18: Interaction screen 2.

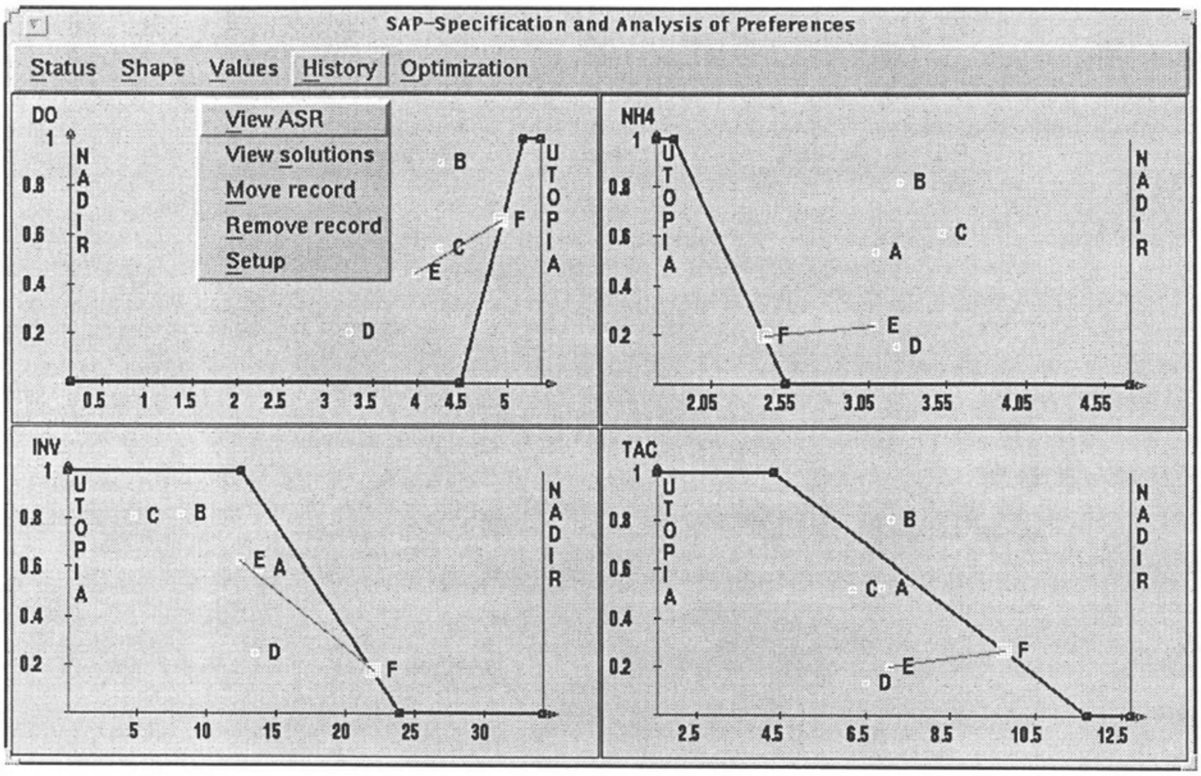

Figure 19: The history in the form of plots.

displayed by choosing the **History** item from the main menu. The first form contains triples for each criterion composed of the criterion value and the values of aspiration and reservation levels. This information (available by the `View ASR` from the `History` menu) usually does not completely fit on a screen and therefore the spreadsheet has to be horizontally scrolled. Therefore another form (composed of

View Solutions

	sol. nr.	DO	NH4	INV	TAC
A	0	2.885923	3.098945	13.8	6.890943
B	1	4.252989	3.257073	8.2	7.103169
C	2	4.232047	3.520948	4.7	6.19206
D	3	3.231534	3.230792	13.5	6.495705
E	4	3.982716	3.089756	12.5	7.068245
F	5	4.942418	2.392354	22	9.774112

Figure 20: The history in the form of spreadsheet.

only criteria values, see Figure 20) is available via the `View solutions` item from the `History` menu of SAP.

14. The specification of the aspiration and reservation levels with the help of the mouse does not result in setting precise values. This is usually acceptable but the user can also set precisely those values by selecting the `Values` item from the SAP main menu.

15. More advanced users may want to specify (in addition to the pairs of aspiration and reservation levels) the piece-wise linear component achievement function. Two examples of such functions are demonstrated on Figure 21. The definition of such

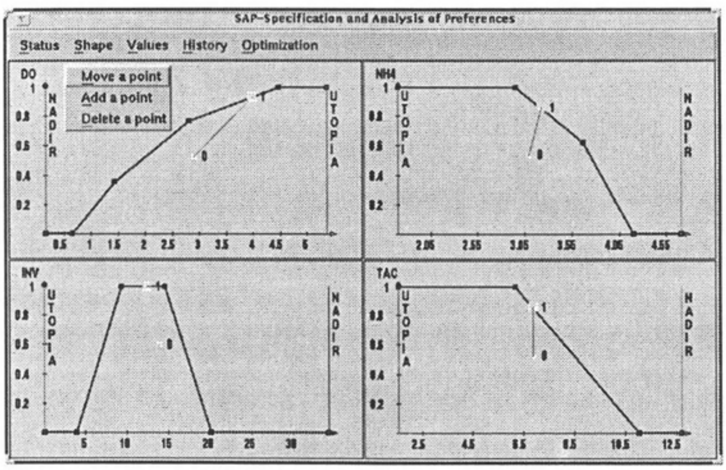

Figure 21: Shape menu and PWL function.

functions require the selection of the `Shape` menu and then the `Add a point` item from this menu. Then each click of the mouse results in adding one point to the component achievement function. Once enough points have been added the `Move a` point should be selected from the Shape menu, which results in switching back to the default mode of SAP (in which a click of the mouse results in moving the nearest point to the point currently pointed at by the mouse. Points no longer needed for the definition of the piece-wise linear functions can be removed by (temporarily) switching the mode by selection of the Delete a point from the Shape menu.

The analysis of the solution can be stopped at any time when the SAP main window is active (it is not active, when the optimization is running) and it can be continued after the next start of the DSS shell. To continue the analysis one should replace selection of New in the first step of the described above interaction by selection of the Open item from the Problem main menu item. The history of the interaction process is saved automatically.

We have selected the Nitra Case Study example as a tutorial for the DSS shell. Once the user becomes familiar with SAP, he/she may want to perform a more realistic analysis of this model. Therefore, we describe below the outline of the problem (see [MSW95a] for details). We consider a river basin or a larger region composed of several basins where the water quality is extremely poor. We also consider a set of waste water treatment plants (either existing or to be possibly constructed) and at each plant, technology (which may be composed of a set of technologies to be selected out of the given set of possible technologies) which can be implemented in order to improve the water quality in a region. The traditional, optimization based approach to solving such a problem consists of looking for a set of plants and technologies whose implementation would result in maintaining prescribed water quality standards at the minimum costs. However an application of such an approach would in this case, as in many other cases, result in an infeasible solution because of the amount of costs involved. Therefore multiple-criteria model analysis has been applied to analyze the trade-offs between three types of costs: investment (INV), operation and maintenance costs (OMRC) and total annual cost (TAC), and the water quality which is represented by three indices: dissolved oxygen (DO), concentrations of CBOD and ammonia (NH4).

A problem specific model generator creates the nitra.cor file (in the LP-DIT format). This file contains the description of the core model, which relates selection of treatment technologies (which are the decision variables) with the resulting costs, waste water emissions, and the resulting ambient water quality. The core model only includes the physical and logical relations therefore there are no built-in constraints for costs nor for the water quality indices. In order to perform the analysis described in [MSW95a] one should also define (in addition to the four criteria suggested for the tutorial example, see Figure 12) the CBOD and OMCR criteria. These are defined by the variables cr_1 and tot_omc, respectively.

The results of the examination of the second core model (AEZ) provided with the mcma are discussed in [AFM95]. The names of the variables defining the criteria are composed of the letter V followed by six (or five) zeros and one (or two for the criterion number 10) digit(s) that correspond to the criterion number. For example, the variable V0000001 defines the criterion Average Food (which has assigned the number 1) whereas the variable V0000010 defines the criterion Maximal Errosion (number 10 on the list of criteria, see [AFM95] for the list and interpretation of the all examined criteria).