IIASA

International Institute for Applied Systems Analysis • A-2361 Laxenburg • Austria
Tel: +43 2236 807 • Fax: +43 2236 71313 • E-mail: info@iiasa.ac.at • Web: www.iiasa.ac.at

INTERIM REPORT IR-98-052 / November

ISAAP – Interactive Specification and Analysis of Aspiration-Based Preferences

Janusz Granat (janusz@ia.pw.edu.pl)
Marek Makowski (marek@iiasa.ac.at)

Approved by
Gordon MacDonald (macdon@iiasa.ac.at)
Director, *IIASA*

Foreword

An important part of a model based decision support is the process of decision selection. This process can be supported by Multi-Criteria Model Analysis (MCMA). Most methods for MCMA 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 deals with the Aspiration-Reservation based approach to the specification of such preferences and with a graphical interface for interactive multi-criteria model analysis.

A modular software tool which supports this approach has been developed within the collaboration between the Risk, Modeling and Policy (formerly 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. It has also been applied to several engineering applications at the Warsaw University of Technology. The presented paper documents the tool and provides a tutorial for its use.

Abstract

Model based Decision Support Systems (DSS) often use multi-criteria optimization for selecting Pareto-optimal 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 the specification of these levels as well as for the 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, which reflect the degree of satisfaction with given values of criteria.

The paper outlines the methodological background and modular structure of a tool (called MCMA) for multi-criteria analysis of decision problems that can be represented as Linear Programming (LP) or Mixed Integer Programming (MIP) problems. The MCMA has been used at IIASA for the analysis of decision problems in water quality management and land use for sustainable development planning. These experiences have shown that the MCMA tool is applicable also to large LP and MIP problems. Other implementations of the same methodology have also been used for the analysis of non-linear problems in several engineering applications.

The current implementation of the modular software tool for interactive specification of user preferences is described in more detail. The presented methodology of multi-criteria model analysis and the documented software is illustrated by a detailed tutorial example.

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

Contents

ISAAP – Interactive Specification and Analysis of Aspiration-Based Preferences

Janusz Granat[*] *(janusz@ia.pw.edu.pl)*
Marek Makowski (marek@iiasa.ac.at)

1 Introduction

A Decision Support System (DSS) is a computerized tool which helps to analyze a decision problem. Model based decision support often uses multi-criteria optimization for selecting Pareto-optimal solutions. Such a selection is based on interactive specification of user preferences. The paper outlines the methodological background and presents a tool that provides a graphical interface for interactive multi-criteria model analysis. The presented tool has been applied to teaching and to several real-world applications which has proved its applicability to both small tutorial and large real-world models.

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* (often referred to as *substantive 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: Selection from the set of all feasible solutions (defined by the *core model*) of a subset of acceptable solutions and then a further selection from such a subset of one solution that corresponds best to the preferences of the DM. Such selections are based on a representation of a preferential structure of a DM. This can be done by a selection of criteria for a multi-criteria based model analysis, or by a selection of one criterion and additional constraints for a single-criterion optimization. Each method of the representation of a preferential structure of a user has a number of parameters that have to be set by the user in order to formulate a corresponding optimization (or simulation) problem. The analysis is often done in an interactive way, thus allowing a

[*]Institute of Control and Computing Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw *and* Institute of Telecommunications, Szachowa 1, 04-894 Warsaw, Poland.

user analysis of previously obtained solutions, changing the representation of his/her preferential structure thus formulating a corresponding underlying optimization problem.

Computations: Model analysis requires solution of a series of auxiliary optimization or simulation problems, which in turn requires a robust and efficient solver that can handle the related computational tasks in a way that is transparent for a user of a DSS.

This paper is an updated version of the documentation of the predecessor of ISAAP documented by Granat and Makowski (1995), and deals with an extension of the aspiration-led multi-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 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 ISAAP, 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 ISAAP is an abbreviation of the full name **I**nteractive **S**pecification and **A**nalysis of **A**spiration-based **P**references, which characterizes 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 the MCMA (Multi-Criteria Model Analysis) software, which can be used for analysis of decision problems represented by an LP or a MIP problem and presents functions of ISAAP. Section 4 provides the information needed for using MCMA and ISAAP by the user who wants either to examine the tutorial examples distributed with this software or to use MCMA and ISAAP to analyze another model. Section 5 contains a tutorial session which illustrates the use of MCMA and ISAAP for analysis of one of the tutorial models. Section 6 informs about the availability of the software and documentation.

2 Methodological background

Discussion on different approaches to decision making support is clearly beyond the scope of this paper. A large bibliography can be found e.g. in (Eschenauer, Koski and Osyczka, 1990; Korhonen, Lewandowski and Wallenius, 1991; Lewandowski and Wierzbicki, 1989b; Makowski, 1994a; Stewart, 1992; Wessels and Wierzbicki, 1993; Makowski, 1996). We will deal with one of the most successful classes of DSSs (see e.g. (Korhonen and Wallenius, 1989) for a justification of this statement), namely with model based DSS which uses aspiration-led multi-criteria optimization as a tool for computing and selecting efficient solutions. This approach, originally proposed in (Wierzbicki, 1980; Wierzbicki, 1982), 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 (Wierzbicki, 1980; Lewandowski and Wierzbicki, 1989a). A unified procedure that covers most of these approaches has been proposed in (Gardiner and Steuer, 1994a; Gardiner and Steuer, 1994b).

Aspiration-Reservation Based Decision Support (ARBDS) is an extension of the aspiration-led multi-criteria model analysis. The ARBDS methodology has been implemented in a number of DSSs presented in (Lewandowski and Wierzbicki, 1989b). The relations between ARBDS and other approaches to multi-criteria optimization are discussed in more detail in (Makowski, 1994b). ARBDS can also be considered, as demonstrated by Ogryczak and Lahoda (1992), as an extension of Goal Programming (see e.g. (Charnes and Cooper, 1967) for details), most probably the oldest technique for multi-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 three-stage approach:

- First, a *core model* is specified and generated. The core model contains only 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, objectives, performance indices).
- Second, in the preparatory stage of the MCMA, 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, MCMA automatically 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.
- Third, 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 MCMA to the *core model* thus forming an optimization problem, whose solution results in a Pareto-optimal solution that

[1]Note, that a variable can represent also more complicated forms of criteria (like following a trajectory, 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 (Makowski, 1994b).

[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 (see e.g. (Isermann and Steuer, 1987)) 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.

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 ISAAP handles the interaction with the user in the third 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 status (see Section 2.2.1). 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 by a piece-wise linear CAF. The methodological background for this option is presented in Section 2.3 and its implementation is documented in Section 4.4.1.
2. The DM can change the status of each criterion. The default status can be changed to *inactive*, *disregarded* or back to the original status (which is one of *min, max, goal*, depending on the type of the criterion). This is supported by the *Status* option (see Section 4.4.3).
3. The DM can change the shape of component achievement function corresponding to each criterion by either defining piece-wise linear function (for the criterion values between aspiration and reservation) or by stabilizing a criterion. This is supported by the *Shape* option (see Section 4.4.1 and 4.4.2, respectively).
4. 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 4.5).
5. 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 MCMA (see Figure 50 on page 44).
6. 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 4.3.2. Once the optimization is selected, the MCMA takes control of the program flow, MCMA generates a single-criterion optimization problem whose solution is a Pareto-optimal solution which corresponds to the current 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-optimal 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

Multi-criteria optimization methods typically assume that a multi-criteria 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 (Makowski, 1994b)) in the terms of Achievement Scalarizing Function (ASF). The concept of ASF has been introduced by Wierzbicki (see e.g. (Wierzbicki, 1977; Wierzbicki, 1986; Wierzbicki, 1992; Wierzbicki, Makowski and Wessels, 1999) for the mathematical foundations, interpretations and applications) and it is very useful for comparing different approaches to multi-criteria optimization.

The selection of the Pareto-optimal solution 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 a 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 defined below) smaller than $(1+1/\epsilon)$. For a non-attainable $\bar{q}$ the resulting Pareto-optimal 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-optimal solution is uniformly better. Setting a value of ϵ is itself a trade-off between getting a too restricted set of properly Pareto-optimal solutions or a too wide set practically equivalent to weakly Pareto-optimal 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 (i.e. a set composed of Pareto-optimal solutions). Much less attention is given to the problem of defining the scaling coefficients[4] coefficients w. A detailed discussion on scaling coefficients in a scalarizing function is beyond the scope of this paper. The four commonly used approaches are summarized in (Makowski, 1994b). In practical applications the most promising approach is based on the calculation of scaling coefficients (that are used in definition of the weighted 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 of DSS described in (Lewandowski and Wierzbicki, 1989b).

The ASF for the ARBDS approach usually takes the form:

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

[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 scaling coefficients w should not (see e.g. (Makowski, 1994b; Nakayama, 1994) for a detailed discussion and examples) be used as weights for a conversion of a multi-criteria problem 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.

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 CAF (defined later in detail), which can be simply interpreted as nonlinear monotone transformations of q_i taking into account the information represented by $\bar{q}_i$ 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 MCMA 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 MCMA *an inactive criterion* (see Section 4.4.3). 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 (Makowski, 1994b) for more details). Therefore, the following form of the ASF is implemented in MCMA in order to facilitate a proper handling of inactive criteria:

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

where I and $\bar{I}$ are sets of indices of active and inactive criteria, respectively, 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 (CAF) $u_i(\cdot)$ are strictly monotone (decreasing for minimized and increasing for maximized criteria, respectively) functions of the objective vector component q_i with values

$$u_i(q_i^U, \cdot) = 1 + \bar{\gamma}, \qquad u_i(\bar{q}_i, \cdot) = 1, \qquad u_i(\underline{q}_i, \cdot) = 0, \qquad u_i(q_i^N, \cdot) = -\underline{\gamma} \qquad (4)$$

where $\bar{\gamma}$ and $\underline{\gamma}$, are given positive constants, typically equal to 0.1 and 10, respectively. In MCMA the values of $\bar{\gamma}$ and $\underline{\gamma}$ are computed for each criterion in such a way that the ratios of the slopes[5] of the adjoint segments to the slopes of segments defined by the Utopia and aspiration points, and by the Nadir and reservation points, respectively (see Figure 1 for the illustration) have given values. These values are in the current implementation of MCMA set to 10 and 0.1, for the segments adjoint to the aspiration and to the reservation point, respectively. Such an approach assures a correct handling of cases, in which a user specifies a flat or a steep segment (adjoint to the aspiration and to the reservation point, respectively) of the CAF.

The piece–wise linear component achievement functions (CAF) u_i proposed by Wierzbicki (1986) are defined by (5) and by (6) for minimized and maximized

[5]Defined by eq. (8).

criteria, respectively.

$$u_i(q, \bar{q}, \underline{q}) = \begin{cases} \zeta_i w_i(\bar{q}_i - q_i) + 1, & \text{if} \quad q_i < \bar{q}_i \\ w_i(\bar{q}_i - q_i) + 1, & \text{if} \quad \bar{q}_i \leq q_i \leq \underline{q}_i \\ \eta_i w_i(\underline{q}_i - q_i) & \text{if} \quad \underline{q}_i < q_i \end{cases} \tag{5}$$

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

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 (4).

However, in order to allow for either specification of only aspiration and reservation levels or for additional specification of preferences (for the criteria values between aspiration and reservation levels), the ISAAP supports specification of the component achievement functions in a more general form than that of eq. (5, 6). For this purpose, the piece-wise linear CAF u_i are defined by segments u_{ji}:

$$u_{ji} = \alpha_{ji} q_i + \beta_{ji}, \qquad q_{ji} \leq q_i \leq q_{j+1,i} \qquad j = 1, \ldots, p_i \tag{7}$$

where p_i is a number of segments for i-th criterion. Such a function for a minimized criterion is illustrated in Figure 1. The thin line corresponds to a function that is composed by three segments, which are defined by four points, namely `U`, `A'`, `R` and `N` (that correspond to the Utopia, aspiration, reservation and Nadir points, respectively). The solid line represents a modified function for which the previously defined aspiration level `A'` was moved to the point `A` and two more points `P`^{1} and `P`^{2} were interactively defined.

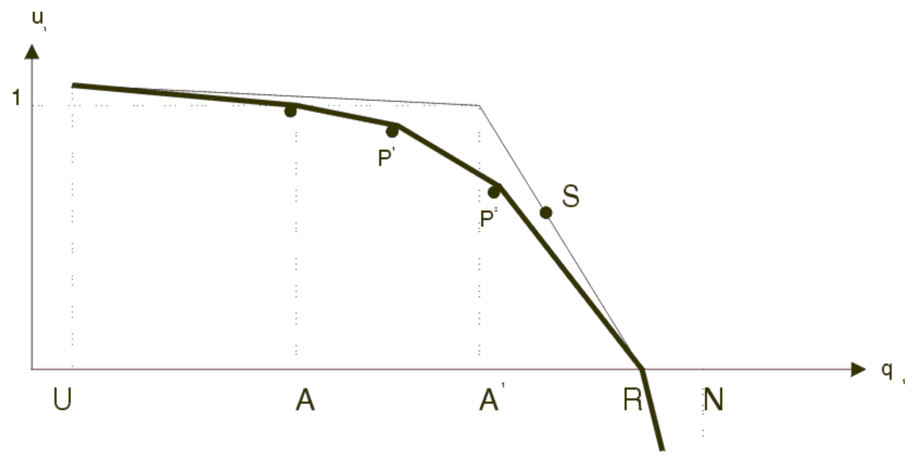


Figure 1: Illustration of the piece-wise linear component achievement function for a minimized criterion.

Practical applications show 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 where an aspiration level is set to the Utopia value and a reservation level is equal to an approximation of Nadir) ISAAP allows for $p_i \geq 1$.

The coefficients defining the segments are given by:

$$\alpha_{ji} = \frac{u_{j+1,i} - u_{ji}}{q_{j+1,i} - q_{ji}} \tag{8}$$

$$\beta_{ji} = u_{ji} - \alpha_{ji} q_{ji} \tag{9}$$

where points (u_{ji}, q_{ij}) are interactively defined with the help of ISAAP (see Section 4.3.1 for details). Concavity of the piece-wise linear functions $u_i(q_i)$ defined by segments (7) can be assured by a condition:

$$\alpha_{1i} > \alpha_{2i} > \ldots > \alpha_{p_i i} \tag{10}$$

Note that the component achievement functions u_i defined by (7) take the same form for minimized and maximized criteria. However, one should add (in addition to the condition (10) that assures concavity) a condition:

$$\alpha_{1i} < 0 \qquad i \in I^{\min} \tag{11}$$

$$\alpha_{p_i i} > 0 \qquad i \in I^{\max} \tag{12}$$

where $I^{\min}$ and $I^{\max}$ are sets of indices of criteria minimized and maximized, respectively. The conditions (11, 12) are fulfilled automatically for the component achievement functions u_i specified with the help of ISAAP.

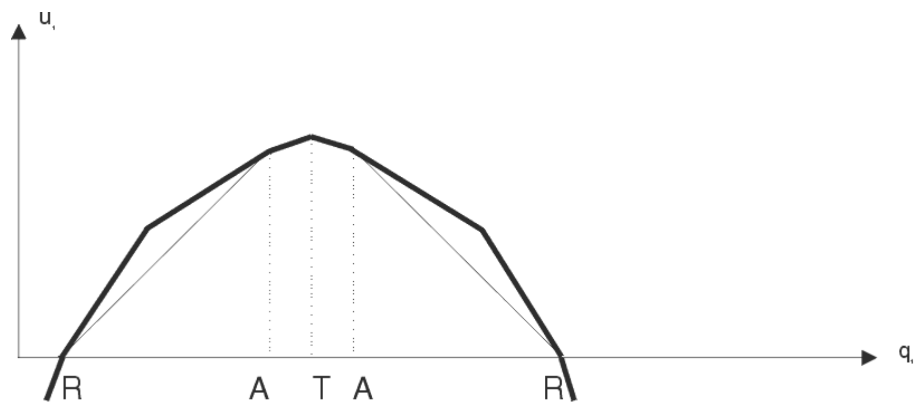


Figure 2: Illustration of the piece-wise linear component achievement function for a goal type (or stabilized) criterion.

A goal-type criterion can be used when a distance from a given target value (which can be changed during the interaction) is to be minimized. For such a criterion a component achievement function is composed of two parts: the first part is defined for the criterion values smaller than the target value, and the second part for the criterion values larger than the given target. Such a function is illustrated in Figure 2. The conditions specified above for maximized and minimized criteria hold for the first and second function, respectively. There is obviously only one point i, for which $\alpha_{i-1,i} > 0$ and $\alpha_{i,i+1} < 0$ and the criterion value for such a point corresponds to a target value (denoted by `T`) for the goal type criterion. The function shown in Figure 2 is symmetric, but for many applications an asymmetric function is appropriate and therefore both types of functions for the goal type of criteria are supported by ISAAP. Figure 35 (on page 37) illustrates an asymmetric CAF.

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 the generation of the component achievement functions (CAF) defined by (5, 6). This is an easy and natural way for reaching attainable Pareto-optimal goals which are defined for each criterion as a pair of aspiration and reservation values. Typically, initial aspiration values are far from being attainable, and the user has to modify her/his preferences (expressed by pairs of aspiration and reservation levels) to decrease a distance between aspiration point and nearest Pareto-optimal solution. The only way to do so is to decrease the aspiration level for at least one criterion. The tutorial section provides a more detailed discussion of this topic, which is crucial for a rational use of most of the multi-criteria model analysis methods.

2.2.1 A type and a status of a criterion

Both ISAAP and MCMA 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 temporarily 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 4.4.3 for details).

2.3 Relations between Fuzzy Sets and Component Achievement Function

In this section we briefly comment on an interpretation of the component achievement functions (5,6) in terms of Fuzzy Sets. 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 (Zimmermann, 1978; Zimmermann, 1985)) 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 allows not only a specification of the user preferences, but also helps him/her in the 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 (Zimmermann, 1987; Sakawa, 1993). These approaches assume that the MF is elicited before the interactive analysis of the problem. Interactive fuzzy multi-objective programming as proposed in (Seo and Sakawa, 1988; Sakawa, 1993) 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 CAF is not elicited at an initial iteration, but the user is allowed to interactively change it upon analysis of obtained solutions. Thus, we are allowing the change of the CAF due to a learning process. The definition of the linear CAF is done by the specification of two points, which are equivalent to the specification of aspiration and reservation levels in the criteria space.

The ARBDS approach uses a so called extended-valued membership function. Such an extension of the membership function concept has been proposed by Granat and Wierzbicki in (Granat and Wierzbicki, 1994), 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 levels, 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 reservation a 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. Such a CAF function is defined in Section 2.1 and is used in MCMA because practical applications show that quite often the user specifies non-attainable reservation levels and/or attainable aspiration levels. In such cases an application of a MF (which has values from the interval [0,1]) as a scalarizing function would result in an optimization problem that has a non-unique solution. In other words, the projection of values of the CAF (or a corresponding extended-valued membership function) on the interval [0,1] can be interpreted as a MF, and such a function is displayed. However, a strictly monotone[6] CAF is actually used in computations in order to provide as a solution a properly Pareto-optimal solution.

The piece-wise linear component achievement functions (7) conform to the requirements for the extended valued membership functions formulated in (Granat and Wierzbicki, 1994). Note that the condition (10) 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 CAF. 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

[6]More exactly, CAF is strictly monotone for minimized and maximized criteria. For goal type criteria and for stabilized criteria, the CAF is composed of two strictly monotone functions (increasing and decreasing) which have one common point (for the criterion value equal to the interactively specified target value).

are between the corresponding aspiration and reservation levels (because the applied CAF 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 terms of degrees of achievements of CAF 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 which degree a solution belongs to a set of *good* solutions. There is no need for restrictions[7] 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 the specification of attainable reservation levels might be difficult.

3 Structure and functions of MCMA

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, a typical model based DSS is composed of a number of mutually linked modules (see (Makowski, 1994b) for a more detailed discussion).

This Section provides an outline of the structure of MCMA, which has been developed for the analysis of LP and MIP (Linear Programming and Mixed Integer Programming, respectively) models using the ARBDS. A typical configuration of a DSS is illustrated in Figure 3 for the application of the MCMA to the Land Use for Sustainable Agricultural Development Planning described and documented in (Antoine, Fischer and Makowski, 1997; Fischer, Granat and Makowski, 1998), respectively. One should note that the structure of the DSS developed for the Regional Water Quality Management in the Nitra River Basin (see (Makowski, Somlyódy and Watkins, 1995; Makowski, Somlyódy and Watkins, 1996) for details) differs only by the type of optimization problem (MIP instead of LP) and the solver used.

The MCMA 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 core model generator (such as documented in (Fischer et al., 1998) and in (Makowski et al., 1995), for land use and water quality management, respectively) for generating the core model, which relates the decision variables with the resulting outcome variables. It is important to stress that the core model

[7]For practical reasons the ISAAP 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.

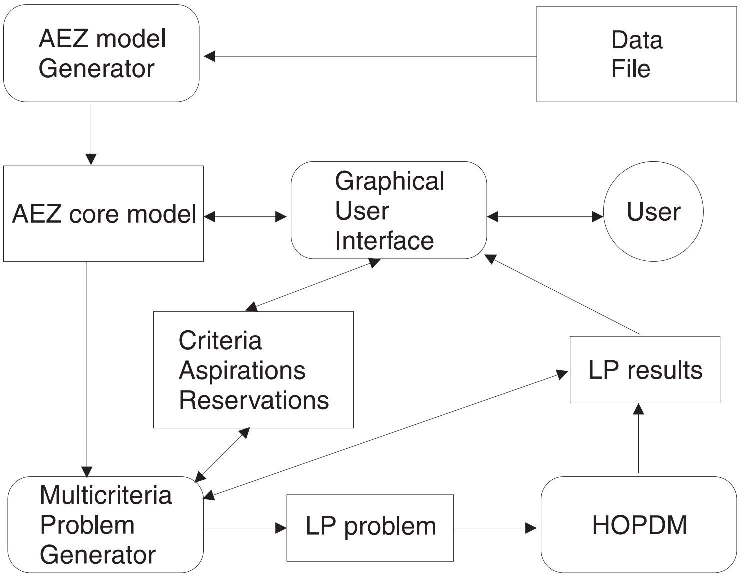


Figure 3: The structure of a Decision Support System for the Land Use for Sustainable Agricultural Development Planning.

includes only physical and logical relations, and not the preferential structure of the DM. A more detailed discussion on core model specification is provided in (Makowski, 1994b).

- The ISAAP described in this paper which supports specification of user preferences both in terms of aspiration/reservation levels and, optionally, more precise specification in terms of piece-wise linear component achievement functions. ISAAP 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 3 the ISAAP is used for the definition of `aspirations, reservations` and for changing the status of `criteria`. However, the ISAAP provides more functions than can be outlined in Figure 3.
- The LP-Multi (see (Makowski, 1994b) for details), a modular tool for handling multi-criteria problems using the methodology outlined in Section 2. An instance of the multi-criteria problem is solved by optimization of a parametric Linear Programming (LP) or Mixed Integer Programming (MIP)[8] problem, which is implicitely defined by the core model and the aspiration and reservation levels, which represent the current preferential structure of a DM.
- Two solvers for solving LP and MIP optimization problems, respectively. The solvers are robust because in a typical application their use is hidden from the user. Therefore solvers used in a DSS must not require interaction with the user. In MCMA an appropriate solver is selected automatically depending on the type (LP or MIP) of the core model. The HOPDM, interior point based LP solver, described by Gondzio and Makowski (1995), is used for LP models while MOMIP, documented by Ogryczak and Zorychta (1996), is applied to MIP problems.
- A data interchange tool lpdit described in (Makowski, 1998). This tool provides an

[8]The type of the parametric problem is the same as that of the core model.

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. The LP-DIT format is used for the definitions of the core model and the LP or MIP problems (the latter defined for each multi-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 (Inm, 1995) and the `zApp` Interface Pack (Inm, 1993). 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.

ISAAP is a module that can be used as a part of a model based DSS using a multi-criteria optimization with aspiration and reservation levels. ISAAP 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 multi-criteria optimization problem, namely:

- Specification of an aspiration and a reservation level.
- Optional specification of a piece-wise linear CAF 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 the analysis of previously computed solutions by handling the solution's history.

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

ISAAP has also been applied as a part of other Decision Support Systems to case studies that require analysis of nonlinear models (see Section 7) without changing a single line of code. This illustrates well the power of modular tools.

4 User guide to MCMA and ISAAP

Currently the ISAAP module is available only as a part of the MCMA. Therefore, this Section provides the following five groups of information (assembled in the corresponding subsections) that are of interest to any user of ISAAP and that correspond to stages of a model preparation and analysis:

- Preparation of a core model,
- The preparatory stage of a model analysis,
- Information about the interaction with ISAAP, which is useful for the user, who is not interested in technical details of the implementation of ISAAP,
- The general implementation assumptions,

• The documentation of files generated by MCMA and ISAAP.

4.1 Preparation of a core model

As already discussed in Section 2, the first stage of the problem analysis is building a corresponding *core model*. Two core models are distributed with this software, one of LP and one of MIP type. These models are taken from the real-world applications, namely:

- Optimizing Land Use for Sustainable Agricultural Development Planning documented in (Antoine, Fischer and Makowski, 1996; Fischer et al., 1998).
- Regional Water Quality Management Problem, case study of the Nitra River Basin (Slovakia) documented in (Makowski et al., 1995; Makowski et al., 1996).

The first of these models is used in the tutorial contained in Section 5. However, it should be stressed that MCMA supports neither a specification nor verification or modification of a 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 (Makowski, 1994b). The preparation of a core model for a non-trivial problem requires not only knowledge and experience, but also software tools. Such tools can be either a problem specific core model generator or a general purpose modeling tool, like AMPL (Fourer, Gay and Kernighan, 1996) or GAMS (Brooke, Kendrick and Meeraus, 1992).

The core model to be analyzed by MCMA and ISAAP must be provided in the form of LP-DIT binary file `id.cor`, where `id` stands for the implied problem name. The tutorial models distributed with this software are already in the LP-DIT format. For LP and MIP type models many of the modeling tools can generate the model in the MPS format. Models prepared by the user in the form of MPS format file can be converted by the LP-DIT utility available from the authors.

4.2 Preparatory stage of a model analysis

In order to allow for a Multi-Criteria Model Analysis (MCMA) ISAAP has to be used within an application, typically composed of a number of modules. In this paper we illustrate the use of ISAAP using as an example the MCMA outlined in Section 3. The corresponding program is called MCMA (from Multi-Criteria Model Analysis). Its main menu is presented[9] in Figure 4, and a tutorial for its use is given in Section 5.

The main menu of this application is reproduced. At this point only three menu items have active[10] submenu items: `Problem`, `Exit` and `Help`.

The `Problem` menu contains the following submenu items that control the corresponding actions:

`New problem` – starts the analysis of a 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

[9]Please see Section 5 for all Figures.

[10]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.

Table 1: Contents of the file with predefined criteria for the AEZ core model.

```
V0000001 var FoodAv   max   Gcal
V0000002 var NetRev   max   mln_KSh
V0000003 var ProCos   min   mln_KSh
V0000004 var GrosOu   max   mln_KSh
V0000005 var Land     min   ha
V0000006 var HarvAr   min   ha
V0000007 var FoodMi   max   Gcal
V0000008 var TotEro   min   tons
V0000009 var SSR      max   0.125%
V0000010 var MaxEro   min   tons/ha
```

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 4.7 for details).

`Continue a session` – 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 the continuation of the analysis done earlier. In order to continue the analysis of the problem, the corresponding 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.

The next step (forced by MCMA) is the definition of criteria which is done interactively (see Section 4.2.1).

4.2.1 Specification of criteria

A criterion is defined by the variable of a core model and the type. Criteria can be specified in a text file and/or can be defined interactively. A core model definition typically uses meaningless names of variables, therefore criteria defined for ISAAP should use meaningful names (instead of using names of the core model variables as criteria names).

An example of specification of criteria is provided in the file aez_dos.cri (its contents is shown in Table 1) that predefines the criteria using the corresponding variables of the AEZ core model. The first word in a line contains the name of a variable (column) or of a constraint (row) of the LP model. The second word must start with either `v` (to indicate that the name corresponds to a variable), or with `c` (for a constraint). Only the first letter of the second word is processed. The third word defines the name of a criterion. The fourth word defines the type of a criterion (one of: min, max or goal). The fifth word defines units in which the respective criterion value is expressed.

Criteria can also be specified interactively. In order to facilitate the 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 have to be composed of printable characters which do not include blank spaces.

The current implementation of MCMA and ISAAP allows for three types of criteria: `minimized, maximized and goal`. The meaning of the 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 neither be minimized nor maximized, but it should attain a given goal (target) value. In many situations this type is useful (for example for the analysis of soft equality constraints) and it has a clear interpretation. For the `goal` type of a criterion the distance between a given target value (which can be changed during the interaction) and a criterion value is minimized. Note that the shape 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 `goal` type criteria 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 a 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 less experienced users.

4.2.2 Computation of pay-off table and compromise solution

After criteria are defined, the MCMA 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 continuation of a previously analyzed problem has been requested, the user can start ISAAP 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 `mcma` 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.

4.2.3 Information stored for further analysis

Quite often a more detailed (typically 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 9). This can be done for the compromise solution and for any solution obtained during the interaction with ISAAP. 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 4.7). 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.

4.3 Basic interaction with ISAAP

After a preparatory stage, described in Section 4.2, is completed one can select the ISAAP item from the MCMA menu available from the main MCMA menu (see Figure 19). The main ISAAP window (shown in Figure 20 on page 30) provides access to a number of functions that can be selected from the main menu of the ISAAP. The most commonly used (and therefore the default) function is described in this Section while more advanced functions are discussed in Section 4.4.

4.3.1 Specification of aspiration and reservation levels

The default function of ISAAP is a selection of aspiration and reservation levels (that define the corresponding CAF) for each active criterion (Section 4.4.3 describes changing of a criterion status). For each criterion the last specified component achievement function $u_i(q_i)$ is plotted[11]. 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 where the mouse is currently pointing to. 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 ISAAP menu.

[11] More exactly, a projection of values of the $u_i(q_i)$ on the interval [0, 1] is plotted.

Aspiration and reservation levels must not be too close (cf. (Makowski, 1994b) for more detailed information). Should the specified values for those levels for a criterion be close, then either they have to 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 Pareto-optimal solution

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

- The control of the program is turned over to MCMA. The optimization problem is generated and solved. The last solution is processed and converted to the ISAAP data structures.
- The control is transferred back to the user, who can either make a new iteration of MCMA or break the analysis.
- The last solution is added to the History records.
- The values of criteria for the last solution are marked by white circles which are connected by a thin line to the corresponding values of the previous solution (marked by green circles).
- The values of criteria of the previous solution are labeled, if the previous solution is in the set of solutions that are displayed (see Section 4.5).

In the current distribution of MCMA the following two solvers are provided:

- HOPDM (see (Gondzio and Makowski, 1995)), interior point based LP solver, especially efficient for medium and large scale problems.
- MOMIP (see (Ogryczak and Zorychta, 1996)), modular optimizer for Mixed Integer Programming (MIP) problems.

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 become unacceptable. A problem specific tuning of solver parameters may substantially improve the performance of a solver.

4.4 Advanced interaction with ISAAP

The following subsections describe more advanced functions of ISAAP, such as a specification of piece-wise linear component achievement functions, stabilizing criteria and changing a status of criteria.

4.4.1 Specification of preferences by an advanced user

The user can specify his/her preferences for the criteria values between aspiration and reservation levels by specification of a piece-wise linear function (cf page 35 for details). In order to create or delete additional points, a corresponding item (`Add a point` or `Delete a point`, respectively) from the `Shape` main menu (see Figure 27 on page 34) should be selected. The points are added or deleted (depending on the selection made from the `Shape` main menu) by clicking the mouse at the place, where a point should be added/deleted. Adding/deleting points is continued until another selection is made from the `Shape` menu. In order to come back to the moving points mode of the interface one should select the `Move a point` from the same menu. The background of the ISAAP window is changed to green or red for adding and deleting points modes, respectively. Additional points can later be moved by clicking the mouse in the way similar to modification of aspiration and reservation levels (cf page 29 for details).

Note that the concavity conditions, defined on page 8 by (10, 11, 12), can be forced by ISAAP only for criteria that do not have additional points. By moving additional points the user may specify a non-concave function which does not fulfill these conditions. In such a case ISAAP informs the user about the points that cause non-concavity of the function and ask the user to modify the corresponding function in order to make it concave.

4.4.2 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 the `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 corresponds to 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 a 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 `stab_sym` type and the following condition is forced by the way in which the interaction is implemented:

$$\tilde{q} = (\bar{q}^+ + \underline{q}^+)/2. = (\bar{q}^- + \underline{q}^-)/2. \tag{13}$$

The `stab_sym` type is the default type for a stabilized criterion. It can be changed from the `Shape` menu to the `stab` type for a criterion in which the user does not accept the condition (13) and wants to specify aspiration and reservation levels independently for surplus and deficit. The `stab` 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 CAF for both (`stab_sym` and `stab`) types of a stabilized criterion.

The target value for a `goal` or `stabilized` criterion can be changed by the mouse (by changing the interaction mode through selection of the `Move a target` from the `Shape` main menu - in this mode the background of the ISAAP window is changed to the yellow color) or from the keyboard (using the dialog activated by selection of the `Values` item from the main ISAAP menu).

4.4.3 Status of criteria

Figure 40 illustrates the dialog (activated by the `Status` item from the ISAAP 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 4.4.2). This selection is suppressed for a `goal` type of a criterion.

`inactive` – a criterion is temporarily disregarded and its component achievement function is not defined. However, a criterion enters the scalarizing function (3).

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

4.5 History

ISAAP keeps record of all the Pareto-optimal solutions and the corresponding aspiration and reservation levels. The history of all solutions can be examined in the form of a spreadsheet (see Figure 36 for the illustration) that can be displayed by the `History` item from the ISAAP menu. The records are arranged in the following way:

- First, `M` solutions that are displayed in the main ISAAP windows. The number `M` is set by ISAAP to be equal to 10, but this value can be changed by the user (by changing the field labeled as `dn` in the dialog shown in Figure 47, page 43). Each solution has a sequence number that is set by ISAAP 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 ISAAP.
- Second, all the solutions that are currently not displayed in the ISAAP main window, sorted by their sequence numbers.

The selected `M` solutions are displayed in the main ISAAP window (see e.g. Figure 41 for the illustration). The previous solutions that belong to the set of displayed solutions are marked by small squares with the labels (the default labels shown in Figure 36 can be changed as described above). If the last but one solution does not belong to this set, then it is marked by a small green circle. The last solution is always marked by a white circle without a label and it is connected with the previous solution by a thin line.

Solutions can be added to or removed from the set of displayed solutions by clicking on the corresponding label (see Section 5.4.3 for details). 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.

4.6 Implementation details

The design and implementation of ISAAP 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 ISAAP 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* (Inm, 1995) and *zApp Interface Pack* (Inm, 1993) are used for GUI. These libraries are available for a number of operating systems therefore, the ISAAP can easily be ported at a moderate cost of purchasing a *zApp* version for another platform. ISAAP is available for Solaris 2.5 and for MS-Windows'95.

ISAAP has been implemented as a C++ class conforming to the draft ANSI C++ specification. Therefore ISAAP 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 ISAAP with an application written in Fortran should also be possible.

4.7 Files created during the analysis

There is 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)[12].

`mc` – binary file with information used by MCMA.

`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.

`sap` – history file maintained by ISAAP.

`lp` – LP or MIP optimization problem in the 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.

[12]For models available in the MPS format one can use the LP-DIT utility provided with the `mcma` software.

5 Tutorial guide to MCMA and ISAAP

MCMA can be used as a stand-alone program or it can be integrated with more advanced decision support systems e.g. such as AEZWIN documented in (Fischer et al., 1998). For testing MCMA as a stand-alone program two core models, namely AEZ (Land Use Planning Model, cf (Antoine et al., 1996; Fischer et al., 1998)) and WQM (Water Quality Model, cf (Makowski et al., 1995; Makowski et al., 1996) for the documentation) are provided with the distributed software. These models are provided in files `aez_dos.cor` and `nitra_dos.cor`, respectively. An instance of the AEZ model is used for the tutorial session described in this Section. The MCMA can be used for analysis of any LP (including MIP) core model that can be provided as LP-DIT format file (a program called `lpdit` is available for conversion of models available in the MPS format into the LP-DIT format).

5.1 On-line help

This report has been written in LaTeX with using additional commands that have been defined in order to make it possible to automatically prepare electronic versions of this document, which in turn can be viewed by one of the two browsers, namely *Netscape* and *zHelp*. Such an approach has several advantages:

- The on-line help is based on an automatically generated electronic version of the corresponding documentation, therefore it is easy to keep the on-line help consistent with a hard copy version of the documentation.
- The on-line help can be viewed by *Netscape* (which is commonly used on both MS-Windows and Unix installations) and/or by *zHelp* (portable browser which is distributed with this application). Due to the limitations of *zHelp* (which does not support the full implementation of the HTML) the functionality of the corresponding version of the on-line help is slightly limited.
- Additional commands for LaTeX define labels which are automatically associated with corresponding pages of the on-line help. These associations are converted into a dictionary, which is distributed with the on-line help. This makes it possible to implement a context-sensitive help, i.e., controlling loading of appropriate pages by the software. However, the context sensitive help is combined with providing the user with a freedom of reading any part of the electronic version of the documentation.

Figure 4 illustrates the way of activating the on-line help. The subsequently displayed dialog shown in Figure 5 provides a choice between the *Netscape* and the *zHelp* browsers. Note that one can use both browsers (by loading them one after another). The welcome pages of both browsers are shown in Figures 6 and 8, respectively. Additionally the information illustrated in Figure 7 is displayed before the *zHelp* browser is shown, if the context sensitive help is enabled for a particular application.

The welcome pages of each browser contain a summary of information pertaining to the use of a particular browser. The use of both browsers is easy and intuitive and therefore no more details about navigating through the on-line help is provided here.

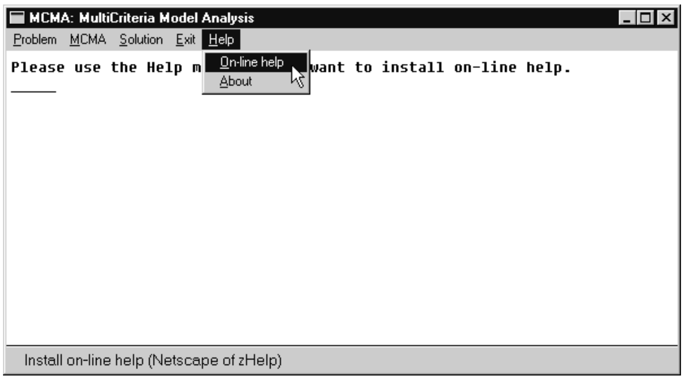


Figure 4: Main menu of the application with the selection of an on-line help.

Figure 5: Dialog for the selection of a help browser.

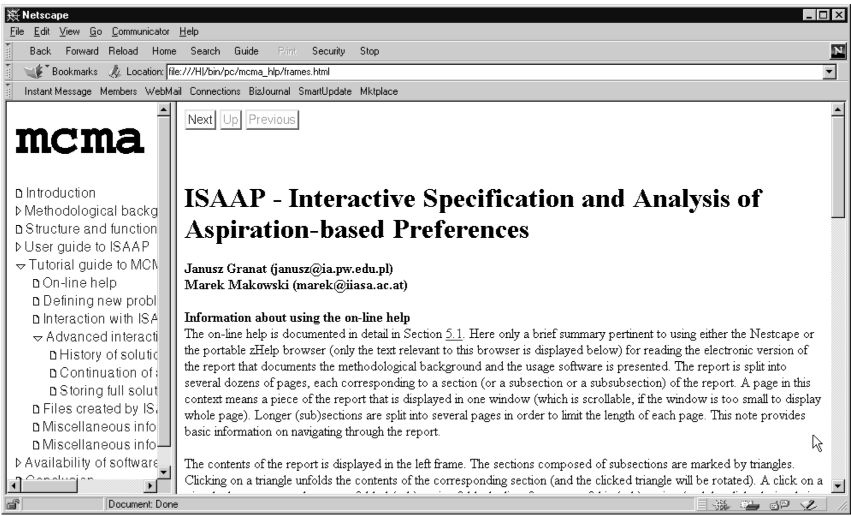


Figure 6: Welcome page of the on-line help viewed by the Netscape browser.

Figure 7: Information about availability of context sensitive help.

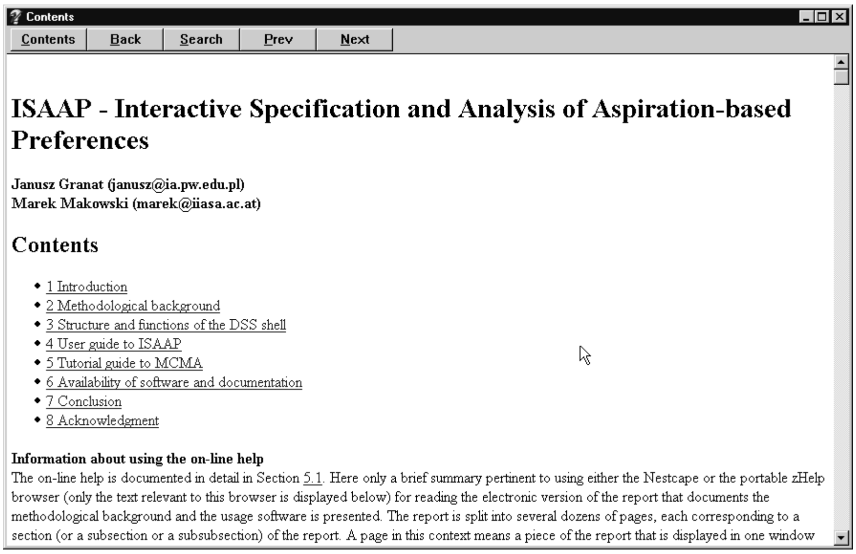


Figure 8: Welcome page of the on-line help viewed by the zHelp browser.

5.2 Defining a new problem

In order to become familiar with the basic functions of MCMA we suggest the following steps:

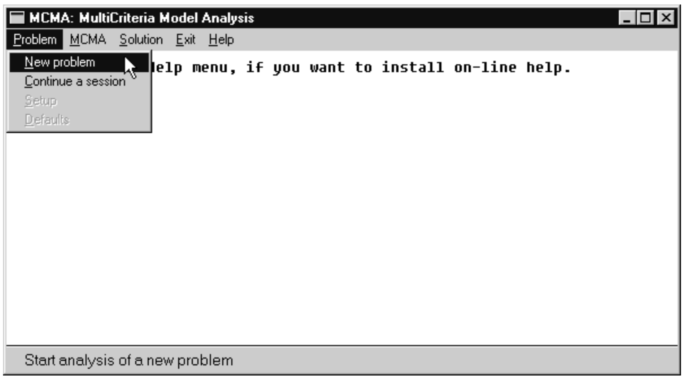


Figure 9: Main menu of the MCMA application (with the selection of the `New problem` item).

- Select the `New problem` menu item (as shown in Figure 9[13]). The list of available core models will be displayed as illustrated in Figure 10. Then select the file name corresponding to the core model you want to examine (in the current distribution the Windows'95/NT version of the AEZ model is called `aez_dos.cor`) and click the mouse on the `OK` button.
- Next, the dialog shown in Figure 11 will be displayed. This dialog allows for a specification of the name of a file containing the predefined criteria (see Section 4.2.1

[13]MCMA run on other operating system will have a slightly different appearance which is determined by a window manager used on such a system. However, the functionality of MCMA will be the same on any operating system.

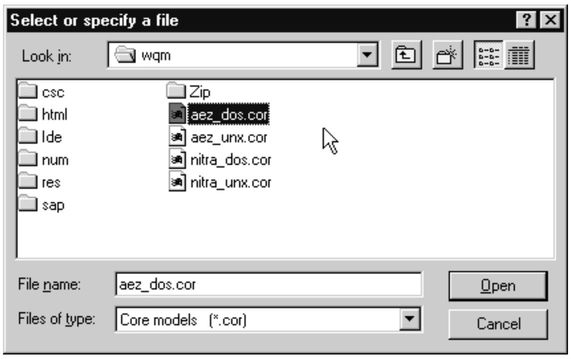


Figure 10: Select a file dialog.

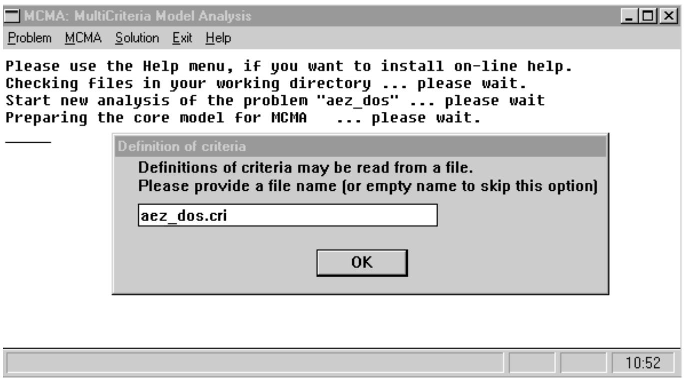


Figure 11: Dialog for selecting a file with predefined criteria (with the main MCMA window in the background).

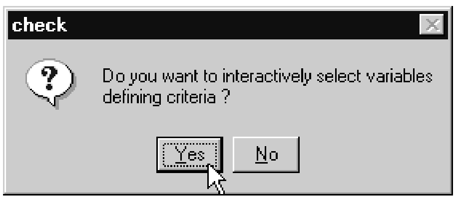


Figure 12: Dialog for activating interactive selection of variables defining criteria.

for details). The default file name is composed of the problem `id` (which is the same as the root of the name of the file containing the core model) and the string `cri` used as the extension of the file name. If the specified file is not readable, then another dialog gives a choice between correcting the name of the file with predefined criteria and refraining from using any file with predefined criteria.

Figure 13: Dialog for selecting a mask for names of variables.

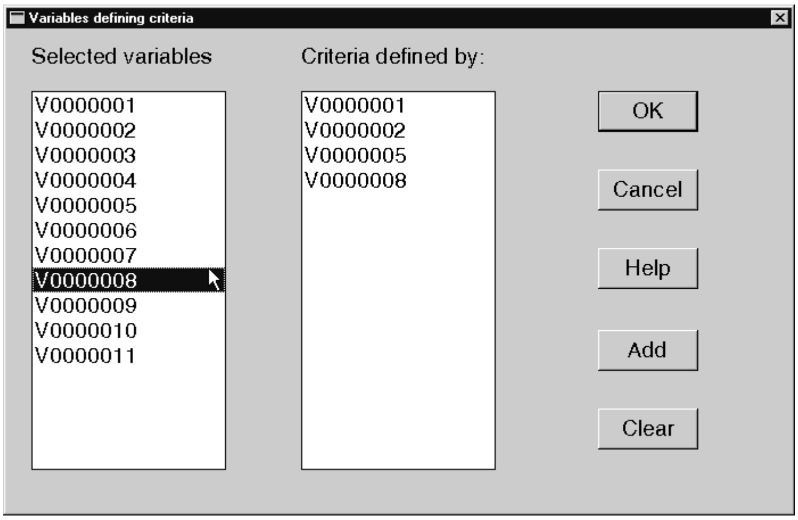


Figure 14: Dialog for selecting variables defining criteria.

- Next, the dialog shown in Figure 12 gives an opportunity of interactive selection of the core model variables, which can be used for specification of criteria (such variables will be added to those used for predefined criteria, if any). If the `Yes` button is pressed for this dialog, then the dialog titled *Select a mask for a name* will be displayed (Figure 13). This gives an opportunity to preselect a group of the core model variables with names that start with a given string (one or more characters). Typically a core model contains hundreds of variables out of which only a few are used as criteria. Selection of an empty string in this dialog would result in providing all variables (which might result in displaying a dialog with several thousands of variables for a large core model).
- In order to select only the variables that are used as criteria in the tutorial, type `V` as a mask name and click the mouse on the `OK` button. The window titled *Variables defining criteria* will be displayed (Figure 14).
- Double click on the name `V0000001` to copy this name from the *Selected variables* list to the *Criteria defined by:* list. Do the same for names `V0000002, V0000005, V0000008`. You may also click on the `Help` button to read more information about this dialog.

- In order to add to the list of variables whose name starts with another string click the mouse on the `Add` button. The dialog titled *Select a mask for a name* will be displayed (Figure 13) again and you can add another group of variables (this is not required for the described example). Click the mouse on the `OK` button when all variables defining the criteria will be selected.
- The next step is to define meaningful names and units for the criteria, to select the type for each criterion and/or to select which predefined criteria will be used for the analysis. This is done by the dialog titled *Definition of criteria* (Figure 15) which is displayed after the selection of variables defining the criteria. For the tutorial example we suggest to ignore all predefined criteria except the following four: `FoodAv, Land, NetRev` and `TotEro`. The names of the criteria which were not predefined are set to `crit_?` (where `?` is replaced by a digit or another character). For such criteria meaningful names and units should be defined. Note that this is not necessary for criteria which were predefined. If one does not use the predefined criteria, we suggest to use the following names for interactively selected variables: `FoodAv` for `V0000001` criterion, `Land` for `V0000002`, `NetRev` for `V0000005`, and `TotEro` for `V0000008`. 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 MCMA for creation of unique names of the parametric optimization problem), and should be composed of ASCII characters (without blank spaces). 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 the `FoodAv` and `NetRev` criteria by clicking on the corresponding radio buttons (the other two criteria will have the default `minimize` type). Double-check the criteria names and types and click the mouse on the `OK` button when you are done.

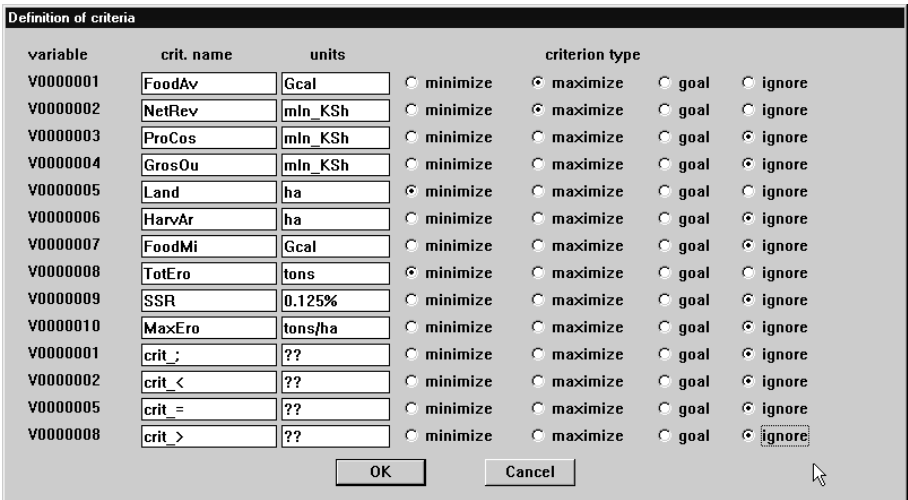


Figure 15: Dialog for definition of criteria (after suggested modifications).

- The next dialog (Figure 16) will ask for the name of the problem. You can define your name or you can use the predefined name by pressing <enter> or by clicking on the `OK` button. Completion of this dialog finishes the preparatory stage of the interaction with the user aimed at the definition of criteria.

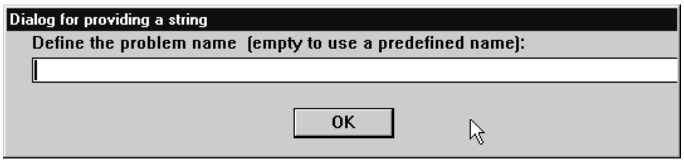


Figure 16: Dialog for definition of problem name.

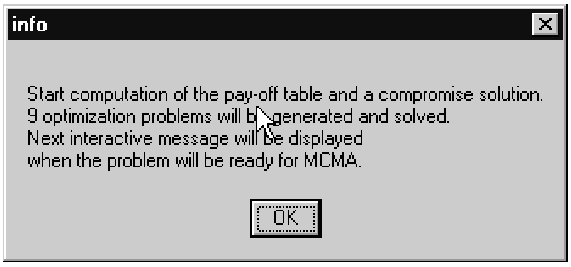


Figure 17: Information about computation of the pay-off table and a compromise solution.

Then the MCMA module starts a series of optimizations in order to compute Utopia point, approximation of Nadir point, and a compromise Pareto-optimal solution. For the n selected criteria this will require solving $(2n+1)$ LP problems each with about 600 variables and 300 rows. The corresponding information will be displayed in a dialog shown in Figure 17. Please click `OK` and wait until the problem will be ready for the interactive analysis. 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 which will be displayed in the main MCMA window. Once the computation of a compromise solution is completed (which on a Pentium-90 PC took for this tutorial example[14] about 3 minutes), the information about the definition of criteria and the pay-off table is stored by MCMA in a file that can be used later for a continuation of analysis. The computation of the compromise solution is reported by a message shown in Figure 18, and the default name of the main window of the MCMA will be changed to the name you selected for the problem. One can at this point store the compromise solution (by selecting the `Store` item from the `Solution` menu) and exit the program.

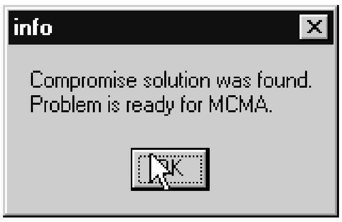


Figure 18: Information about computation of the pay-off table and of a compromise solution.

[14] This example is provided in the `aez_test.cor` file.

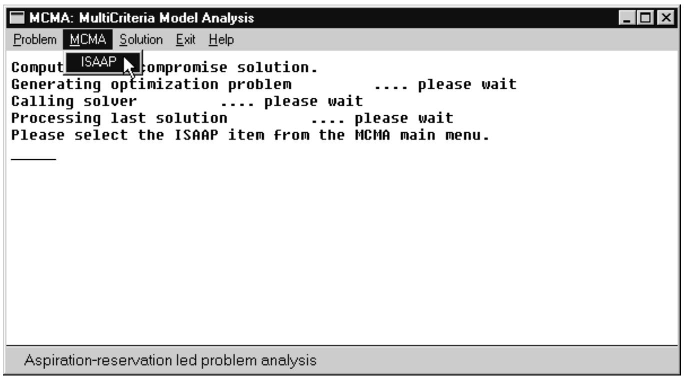


Figure 19: Main MCMA window after computation of the pay-off table and of a compromise solution.

5.3 Basic interaction with ISAAP

After the compromise solution is computed, or after a continuation of the analysis has been selected, ISAAP can be started by selecting the ISAAP item from the MCMA main menu. A compromise solution (such as presented in Figure 20) is typically not acceptable, because the aspiration and reservation levels were set automatically by MCMA to be equal to components of Utopia and Nadir point, respectively. However, such a solution is a good starting point for analysis of the model.

Let's assume that the user wants to primarily improve the level of food production and therefore she/he attempts to achieve this by increasing the reservation level for the corresponding criterion to about 300, without changing aspiration and reservation levels for other criteria (see Figure 21 for the illustration). This can be achieved by a sequence of clicking the mouse to points corresponding to a new reservation level. The readers will probably be surprised why this can not be achieved by a single click. The reason for this originates from the results of experiments on how the interaction should be organized. It seems that the easiest way to move points is by clicking the mouse to a new desired value for the point that is `nearest` to the new desired value. In this case we would like to move the reservation level of the criterion `FoodAv` which was below 100, therefore we should click first to a value around 200, next to around 250, and finally to 300. This description looks complicated, but it is actually easy to become natural.

The Pareto-optimal solution that corresponds to the preferences set as shown in Figure 21 is illustrated on Figure 22. We have indeed succeeded to considerably improve the food production level. However, this resulted in substantial increase of land use and of erosion. Note that the level of net revenue has not changed substantially, but our satisfaction level (measured by the value of the CAF function) has decreased. This decrease corresponds to the fact that the current solution is more distant from the aspiration level than the compromise solution.

Before making further steps in analysing the tutorial model, let us explain a few basic rules applied to the organization of the interaction. Unfortunately, black and white printed figures poorly illustrate some of those rules. However, these figures should help the reader to reproduce the results on her/his color screen.

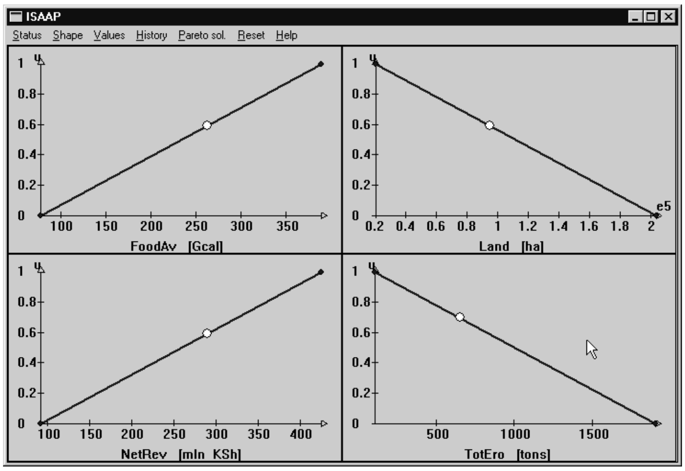


Figure 20: First interaction: compromise solution

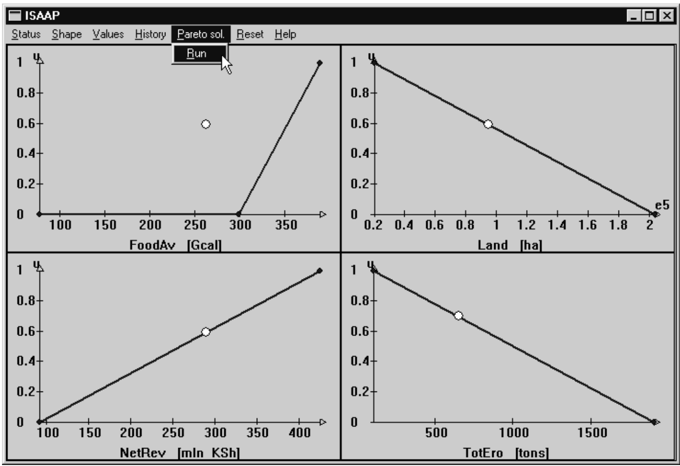


Figure 21: Increasing the reservation level for FoodAv.

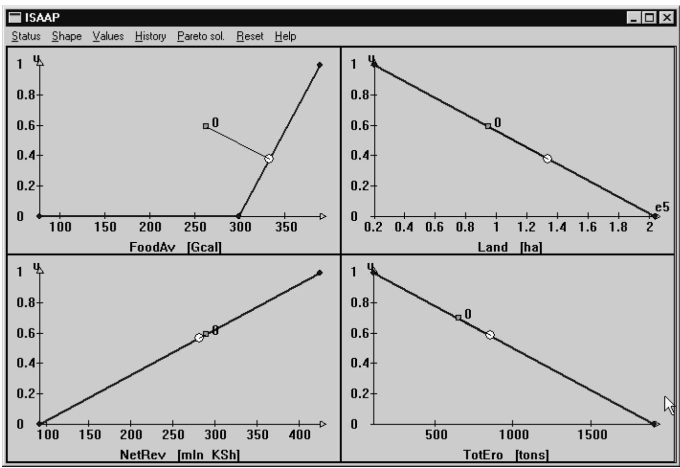


Figure 22: Solution for preferences set in Figure 21.

For each of the defined criteria, the last computed value of each criterion is marked by a small white circle. One iteration consists of the analysis of previous solutions, selection of new aspiration and reservation levels, optional definition of additional segments of piece-wise linear functions defining the component achievement functions, optional change of criteria status, and computation of a new Pareto-optimal solution.

Once you have decided the status of all criteria and the new set of aspiration and reservation levels for each criterion, select the `Pareto sol.` item from the MCMA main menu. This will start generation and solution of the new optimization problem. The ISAAP window is minimized during the optimization and it is restored to its previous size after the new solution is ready for analysis. We will illustrate these options by the following iterations which are commented below.

After the first attempt to improve only one criterion let's assume that we would like now to keep the aspiration and reservation level for the `FoodAv` criterion without changes, and we would like to decrease the erosion. In order to make our current goals more realistic we agree to use more land and we would be satisfied with the current level to the revenue (but we would not like to drop it below 150). Figure 23 illustrates the setting of the aspiration and reservation levels that may represent such preferences. The corresponding Pareto-optimal solution is presented in Figure 24. We can observe that we have achieved only a small improvement for the `TotEro` criterion without much changes of the values of the other three criteria. The decrease of the satisfaction level for the `TotEro` criterion might be surprising because the value of this criterion has improved (the erosion has decreased). However, explanation for this phenomenon is easy, if one remembers the interpretation of the CAF which represents the satisfaction level. In the first (compromise) solution the satisfaction for this criterion was much higher because the reservation level was much higher, therefore the value of this criterion in the compromise solution is meeting our expectations quite well. However, in the second iteration we increased our expectations considerably (by moving the reservation level to the value of 1000), therefore the value of this criterion for the new solution is much less satisfactory (taking into account the new preferences) than in the compromise solution.

By now it is becoming clear that we can not decrease the erosion much further without relaxing our expectations for the other criteria, especially for revenue and food production. Figure 25 illustrates the new settings of aspiration and reservation levels for those two criteria. The preferences for the `Land` criterion are not changed because we may expect that the land use will decrease anyway (since we implicitly agree to decrease the food production). We leave the reservation level for the `TotEro` criterion unchanged (aspiration level is set to the Utopia value, hence it can't be improved) in order to illustrate that one can achieve an improvement of a criterion value without changing aspiration/reservation levels for this criterion.

The Pareto-optimal solution for preferences set as in Figure 25 is illustrated in Figure 26. Note that the value of the `TotEro` criterion has been improved without changing its aspiration and reservation level. This has been achieved by relaxing the aspiration and reservation levels for the `FoodAv` criterion.

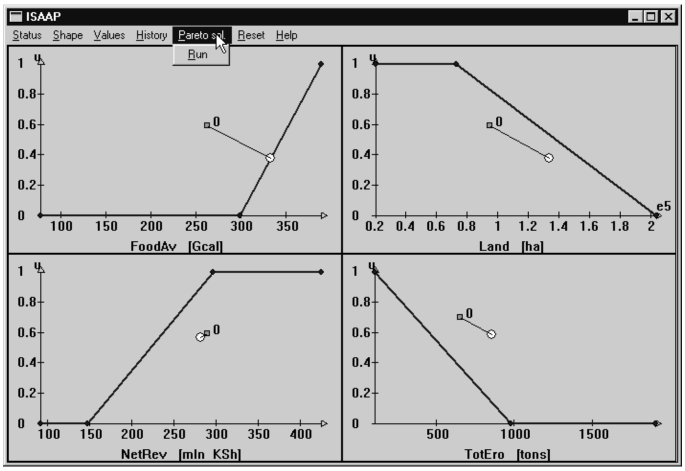


Figure 23: Setting of preferences in the second iteration.

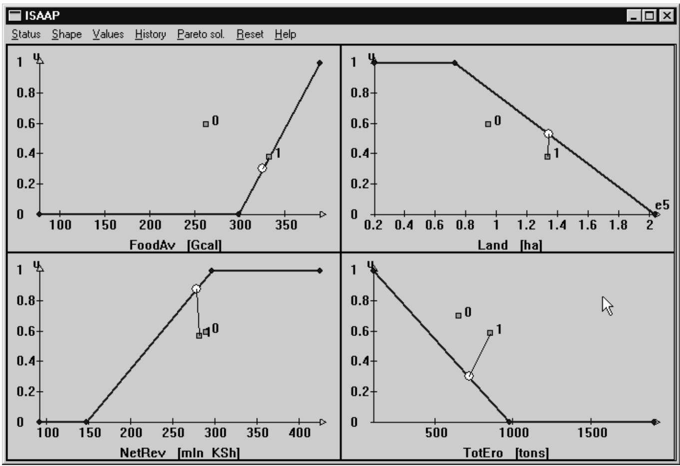


Figure 24: Results of the second iteration.

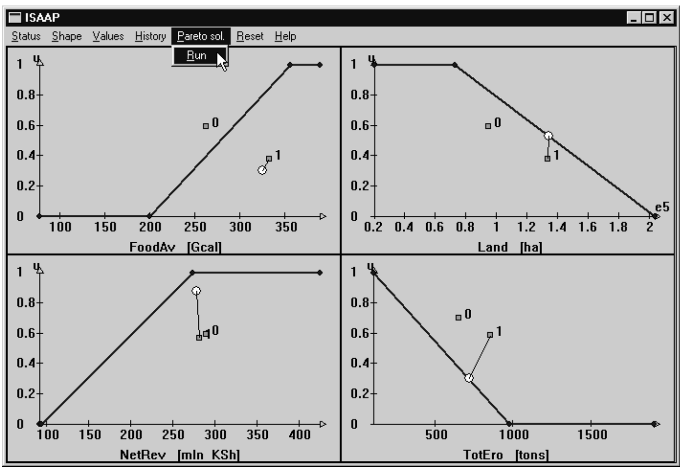


Figure 25: Setting of preferences for third iteration.

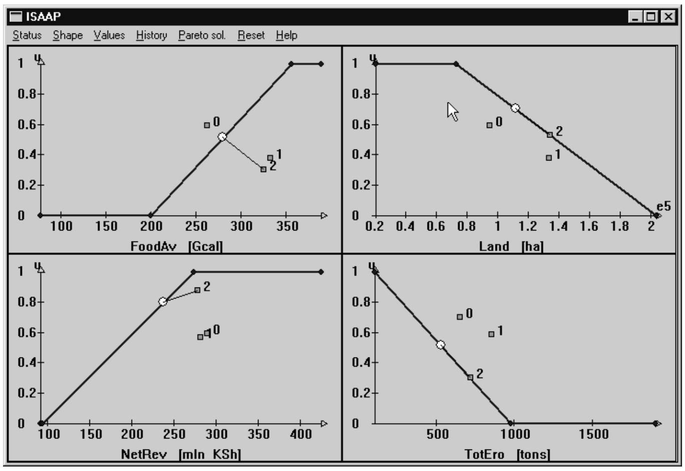


Figure 26: Solution for the third iteration.

5.4 Advanced interaction with ISAAP

Now it is a good time to illustrate other (than just moving aspiration and/or reservation points by clicking a mouse) possibilities of specification of preferences. In every iteration one can use any combination of the following actions for analysis of the problem:

- One can change a shape of a function by selecting the `Change a shape` item from the `Shape` menu (as shown in Figure 27) which activates the dialog illustrated in Figure 28. In this dialog, the `NetRev` criterion has been stabilized. The stabilized criterion has a shape illustrated on Figure 29 (which, of course, can also be changed by clicking the mouse). The dialog presented in Figure 28 provides a selection of two types of a stabilized criterion: `stab` and `stab_sym` which differ by the way in which the shape of the corresponding function can be modified. The `stab_sym` is used for a case, where a symmetric CAF function is desired. In this case, which is a typical one, modifications of the function are done in such a way that symmetry is assured. However, for some criteria a non-symmetric shape may be appropriate. This can be achieved by using the `stab` type.
- The specification of the aspiration and reservation levels with the help of the mouse is very handy, but it does not result in setting precise values. This is usually acceptable, but the user can also set precisely the values of aspiration and reservation levels (and target values for stabilized criteria) by using the `Values` option. This dialog is illustrated in Figures 30 and 31. The first of these Figures contains values set with a mouse, whereas the second one contains values specified from the keyboard through the illustrated dialog. The dialog allows for setting (for each criterion) either a pair of reservation and aspiration values, or a target value for a stabilized criterion. This dialog contains for a minimized or maximized criterion static text with values of Utopia and approximation of Nadir point components (denoted by `U` and `N`, respectively). For any type of a criterion the current value (denoted by `V`) of a criterion is displayed as a static text. The values are set (and the plots are updated accordingly) after clicking on the `Set` button. The dialog has to be closed by the `Close` button.

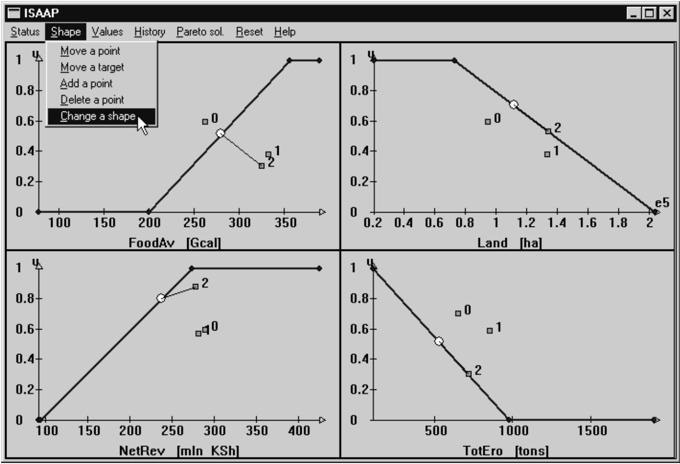


Figure 27: Activation of the dialog for changing shape of a function.

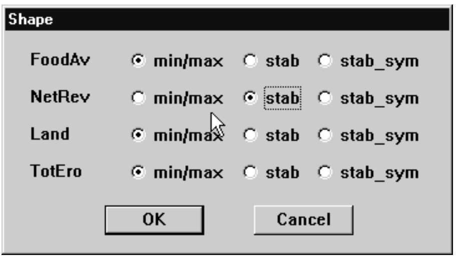


Figure 28: Dialog for changing shape of a function.

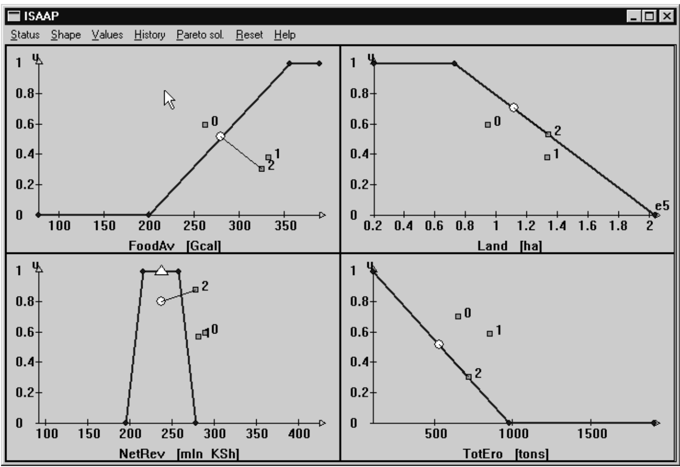


Figure 29: ISAAP window after changing shape of a function.

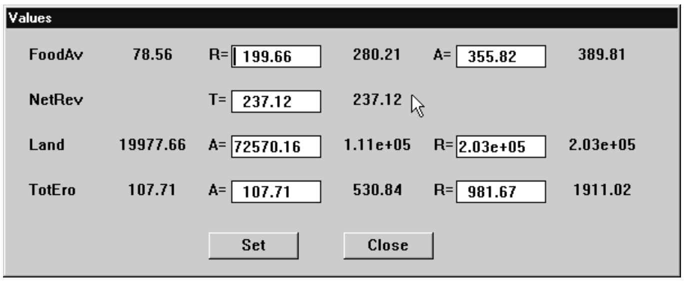


Figure 30: Values set by the mouse displayed in the dialog for setting values from the keyboard.

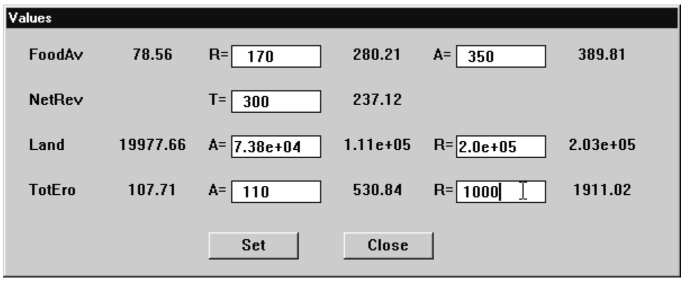


Figure 31: Dialog for setting values from the keyboard after modification of the target, aspiration and reservation values.

- New values of aspiration and reservation levels for any criteria can be set by clicking the mouse near a point which you want to move. However, more advanced users may want to specify (in addition to the pairs of aspiration and reservation levels) the piece-wise linear component achievement function. An example of such a function is demonstrated in Figure 32. One can add more segments to any achievement function by selecting the `Add a point` item from the `Shape` menu. After selecting this option, the background of the ISAAP window will be changed from the default gray color to green and will remain green until one of the following options from this menu will be selected (other options also change the background color to alert a user that he/she should restore the default action of the mouse by selecting the `Move a point` item). In this mode (indicated by the green background) each click of the mouse results in adding (at the current location of the mouse pointer) 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 MCMA (in which a click of the mouse results in moving the nearest point to the place currently pointed at by the mouse). Additional point(s) 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. In this mode (indicated by the red background) each click of the mouse results in removing one point (closest to the current location of the mouse pointer) of the component achievement function.

Figure 32 shows the results of actions corresponding to the three dialogs explained above (selection of stabilized criterion `NetRev`, setting via the dialog target value[15] for this criterion and aspiration and reservation levels for the remaining criteria) and selections of additional segments for the achievement functions for criteria `FoodAv` and `Land`. The changes of the preferences are aimed at the further improvement of the criterion `TotEro`.

The solution corresponding to the preferences set as shown in Figure 32 is illustrated in Figure 33. We have indeed achieved an improvement for the `TotEro` criterion, however, at the cost of considerable decrease of food production.

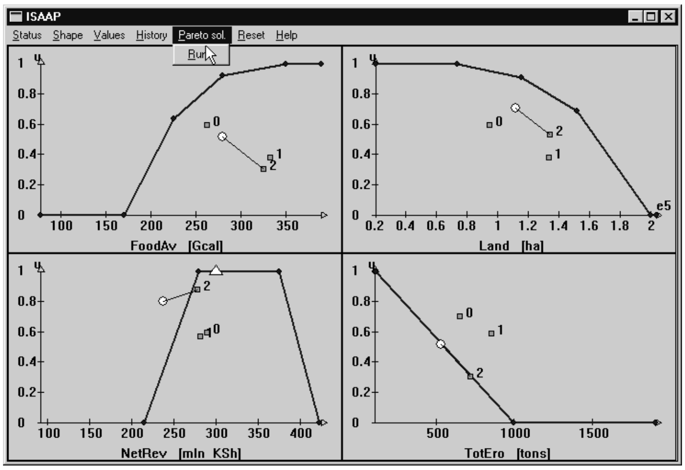


Figure 32: Preferences for fourth iteration.

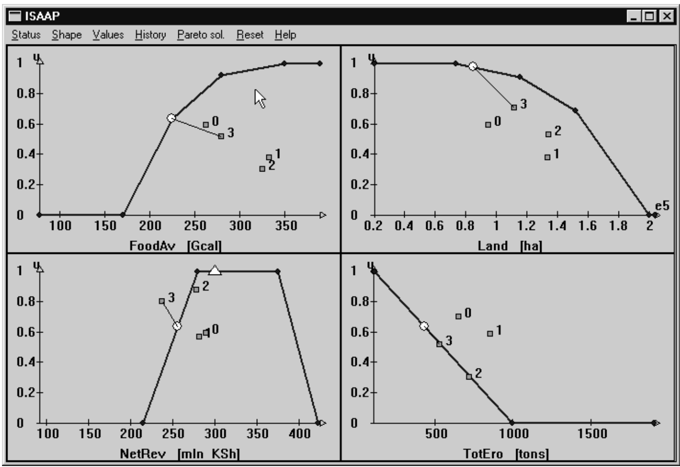


Figure 33: Solution for the fourth iteration.

In our last iteration before the break let us try to increase the food production without increasing erosion. Most probably the only way to achieve this is to relax our expectations for the revenue. Figures 34 and 35 illustrate settings of preferences and the corresponding solution, respectively. Please note that the shape of the component achievement function for the `NetRev` criterion is not symmetric. This

[15]The target value is marked on the graph by a triangle.

corresponds well to the nature of this criterion because one usually would like to have different shapes for values above and below the target value[16]. The last solution shows that it is indeed possible to considerably increase the food production while keeping the erosion level at almost the same level. However, this can be achieved at the expense of a considerable drop of revenue.

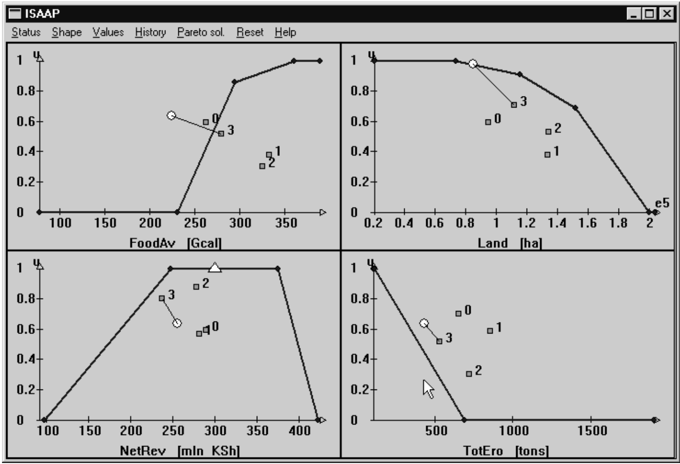


Figure 34: Preferences for the fifth iteration.

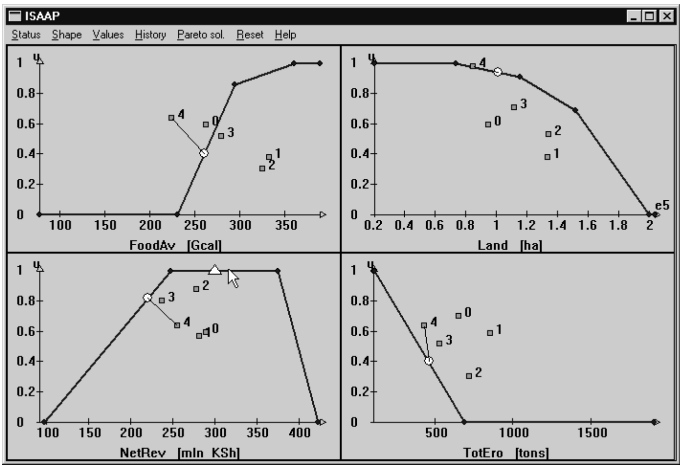


Figure 35: Solution for the fifth iteration.

5.4.1 History of solutions

After several iterations the plots of the achievement functions and the points that mark previous solutions may look like those illustrated in Figure 35. The circles mark the values of criteria corresponding to the new solution and the thin lines

[16]For some engineering applications a symmetric shape of stabilized criterion is often more appropriate.

View Solutions

	sol. nr.	FoodAv	NetRev	Land	TotEro	comment
0	0	262.54	289.57	94990.02	654.92	compromise solution
1	1	333.12	281.57	1.34e+05	853.92	
2	2	326.02	278.40	1.34e+05	717.95	
3	3	280.21	237.12	1.11e+05	530.84	
4	4	225.13	256.14	84584.20	431.30	NetRev stabilized
5	5	260.22	220.81	1.01e+05	460.09	

Figure 36: History in the form of a spreadsheet.

connect the criteria value obtained in the previous and last solutions. The small rectangles in Figure 35 represent the previously calculated solutions. The history of the interactions is provided both graphically and in the form of a spreadsheet. The complete information about the previous optimization run is available in two forms of spreadsheets that can be 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 only criteria values, see Figure 36) is available via the `View solutions` item from the `History` menu of MCMA.

The last field of the history allows for attaching short comments to selected solutions.

5.4.2 Continuation of the analysis

The analysis of a non-trivial problem usually takes several hours and it is typically advisable to break the analysis at some point and continue with it after some time. The analysis of the model can be stopped at any time when the MCMA main window is active (it is not active, when the optimization is running, and the ISAAP window is minimized) and it can be continued after the next start of MCMA. The analysis can be stopped by selecting the `Exit` item from the main menu (cf. Figure 9). The history of the interaction process is saved automatically.

To continue the analysis one should (after starting the MCMA) replace the selection of `New problem` in the first step of the interaction described above by the selection of the `Continue a session` item from the `Problem` main menu item (also illustrated in Figure 9 and then by a selection of a file with the extension `mc`. After selecting the file `aez_dos.mc` the information shown in Figure 37 will be displayed.

If a continuation of analysis is selected, then the files with extension `mc` are shown in the dialog illustrated in Figure 37. Such files contain all information necessary for the continuation of the analysis. The information illustrated in Figure 38 is provided before the continuation of the analysis can be selected.

Figure 39 illustrates the first iteration screen for a continuation of the problem analysis. You may notice that the components achievement functions are set to be defined by Utopia and Nadir point.

We will now illustrate how to change the status of a criterion, which can be

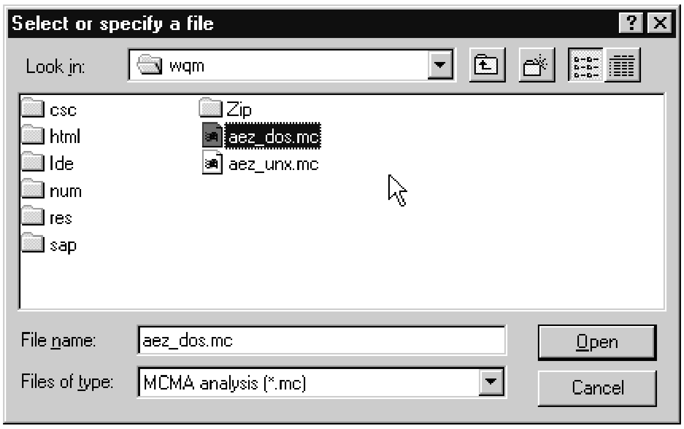


Figure 37: Selection of problem for continuation of the analysis.

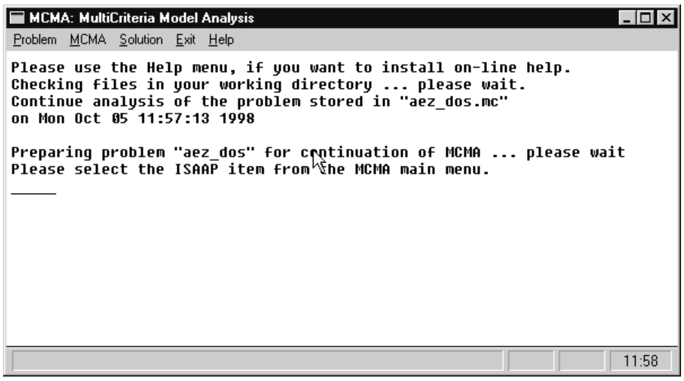


Figure 38: Information for continuation of analysis.

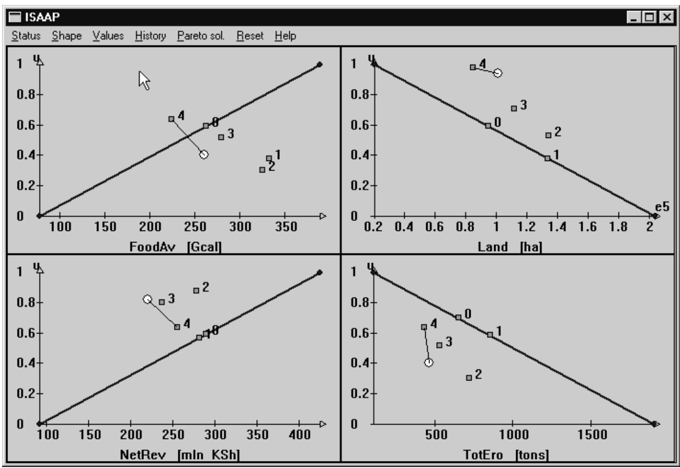


Figure 39: Initial iteration for continuation of analysis.

done by selecting the item `Status` from the main menu. One can make a criterion inactive or disregarded by selecting a corresponding radio box button. Selecting the original type of a criterion (defined in the initial stage to be either minimization, maximization or goal) makes a criterion active again. The dialog for changing criteria status is shown on Figure 40. For the sake of illustration we select the criterion `Land` to be inactive.

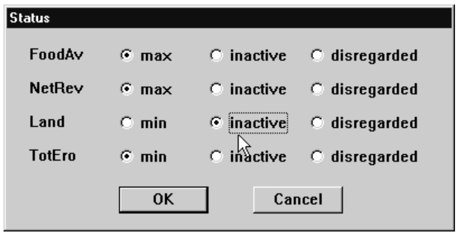


Figure 40: Changing status of criteria.

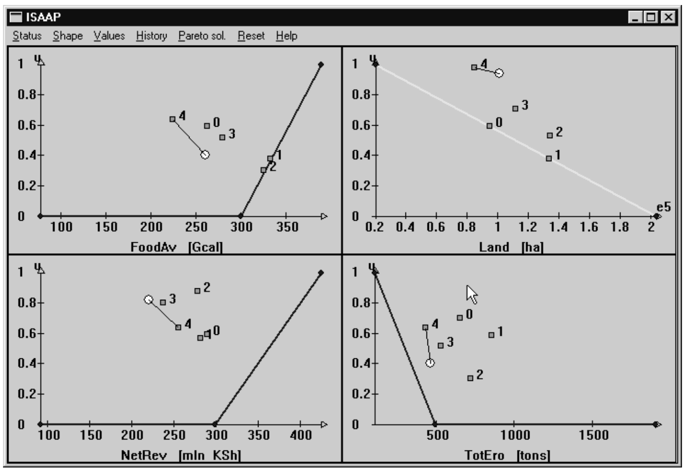


Figure 41: Preferences for sixth iteration.

Further on we will illustrate two attempts to select unrealistically high reservation levels for all criteria. Figure 41 shows a setting of reservation levels for all three active criteria to be close to the Utopia values. The corresponding solution is shown in Figure 42. This solution illustrates well one of the nice features of the ARBDS approach. Namely, instead of reporting an infeasible solution, (which would be the case, if reservation levels would be represented by hard constraints), a Pareto-optimal solution is found with the best possible criteria values which are all in this case worse than the corresponding reservation levels.

Figures 43 and 44 illustrate preferences and solutions, respectively for the seventh iteration, which was an attempt to improve the values of two criteria which had values below the reservation levels by a considerable relaxation of reservation level for the third criterion. The result of this attempt is negative, although the value of the `FoodAv` criterion has been improved, it is still worse than the reservation level, and the value of `TotEro` was slightly worse.

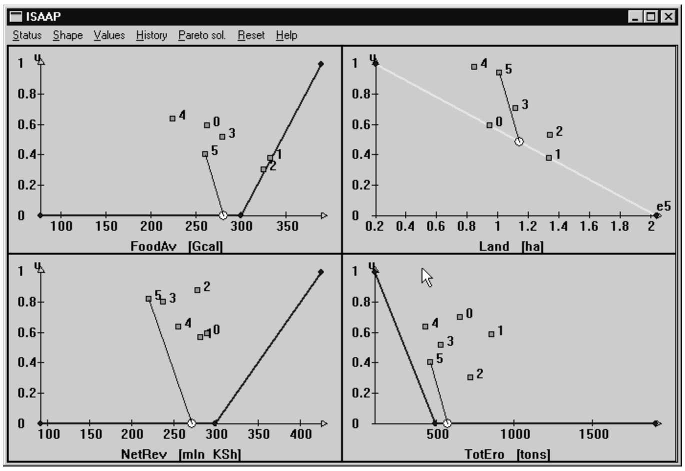


Figure 42: Solution for sixth iteration.

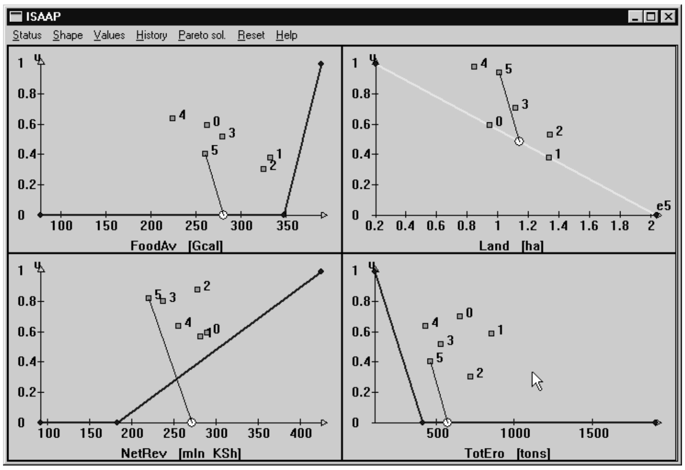


Figure 43: Preferences for seventh iteration.

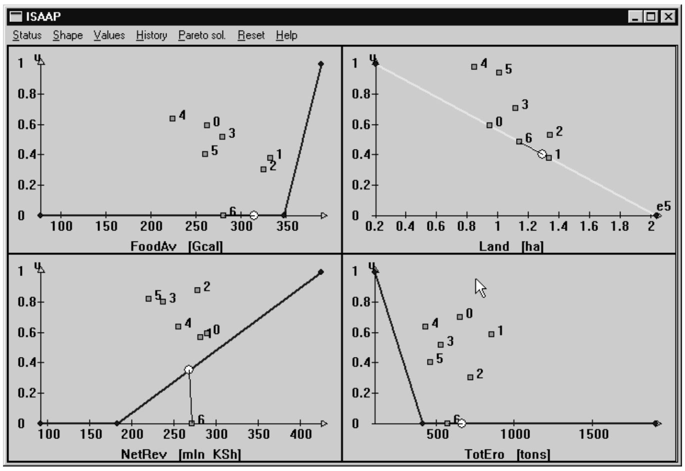


Figure 44: Solution for seventh iteration.

Finally, Figures 45 and 46 show how one can achieve a very good value also for the `TotEro` criterion by relaxing the reservation level for the food production.

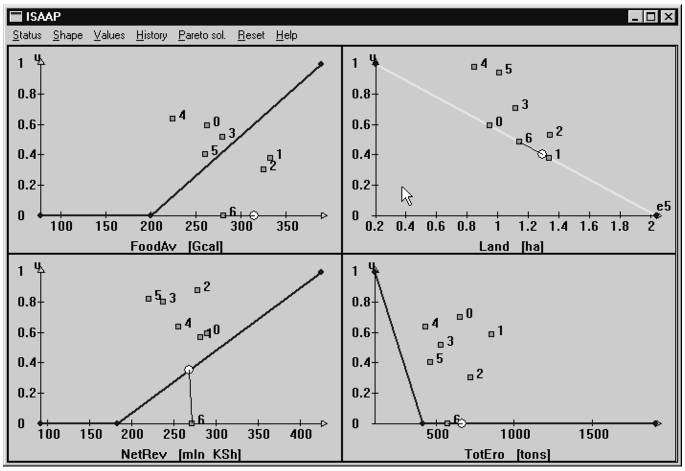


Figure 45: Preferences for eighth iteration.

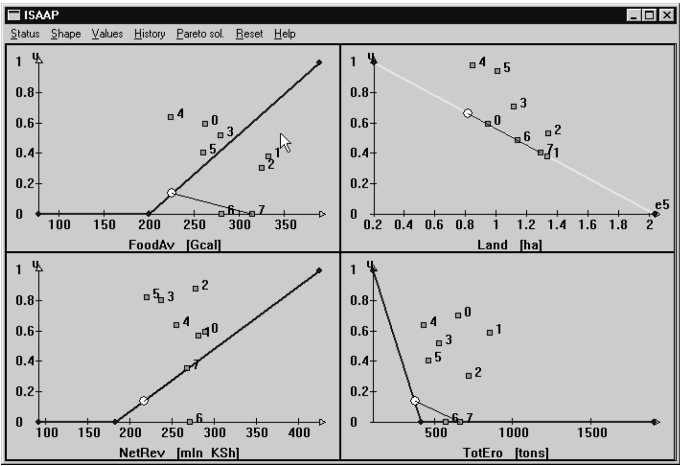


Figure 46: Solution for eighth iteration.

5.4.3 Rearranging the set of displayed solutions

For practical reasons ISAAP displays maximum 10 solutions. This number can be decreased by a specification of a smaller number in the field `dn` of the dialog (that can be activated by selecting the `Setup` item from the `History` main menu item) shown in Figure 47. The same dialog can be used for changing the labels that mark the solutions (from the default digits to other characters).

The first `dn` solutions from the history are displayed. A solution can be added to the set of displayed solutions by selecting the `Move` item from the `History` main menu item. One should select first a solution that should be added to this set by clicking on the label as shown in Figure 48. In this example the solution number 8

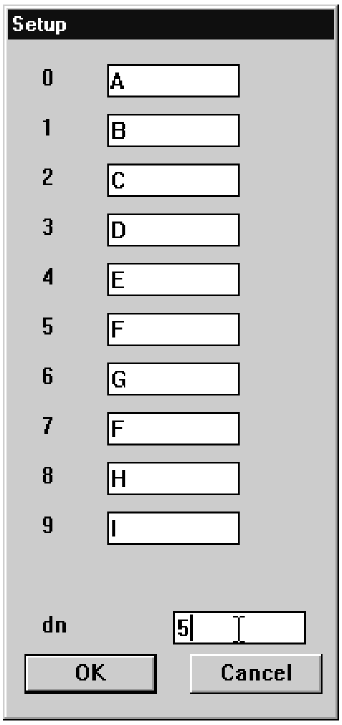
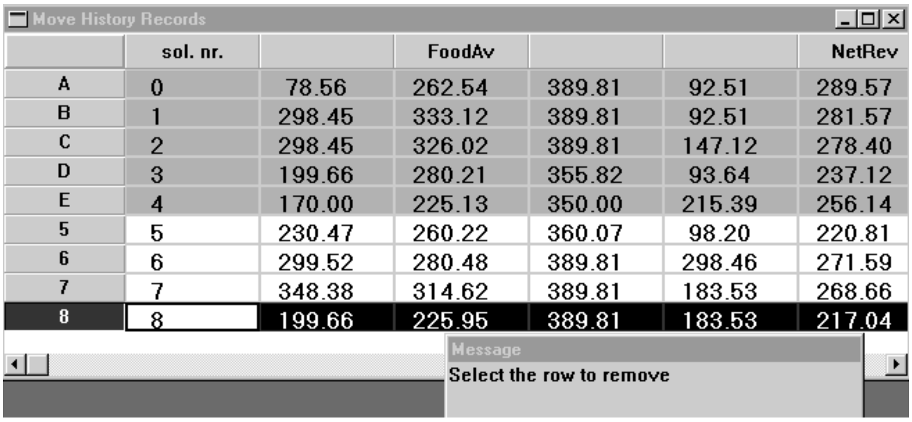
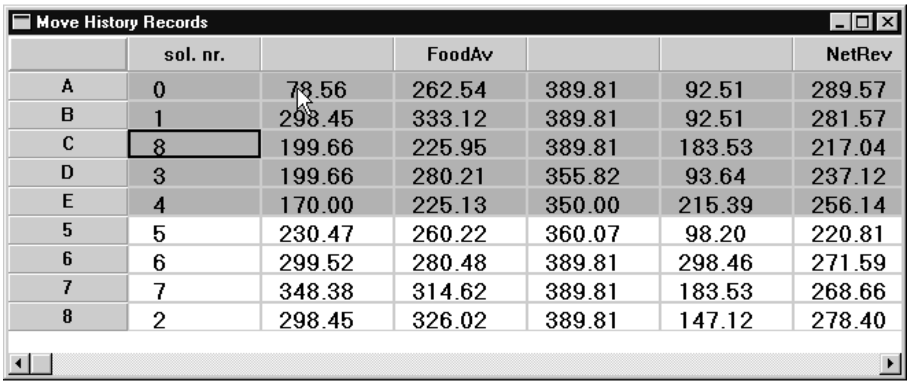


Figure 47: Dialog for changing labels and number of displayed solution.



	sol. nr.		FoodAv			NetRev
A	0	78.56	262.54	389.81	92.51	289.57
B	1	298.45	333.12	389.81	92.51	281.57
C	2	298.45	326.02	389.81	147.12	278.40
D	3	199.66	280.21	355.82	93.64	237.12
E	4	170.00	225.13	350.00	215.39	256.14
5	5	230.47	260.22	360.07	98.20	220.81
6	6	299.52	280.48	389.81	298.46	271.59
7	7	348.38	314.62	389.81	183.53	268.66
8	8	199.66	225.95	389.81	183.53	217.04

Figure 48: Rearranging solutions in the history.



Move History Records

	sol. nr.		FoodAv			NetRev
A	0	78.56	262.54	389.81	92.51	289.57
B	1	298.45	333.12	389.81	92.51	281.57
C	8	199.66	225.95	389.81	183.53	217.04
D	3	199.66	280.21	355.82	93.64	237.12
E	4	170.00	225.13	350.00	215.39	256.14
5	5	230.47	260.22	360.07	98.20	220.81
6	6	299.52	280.48	389.81	298.46	271.59
7	7	348.38	314.62	389.81	183.53	268.66
8	2	298.45	326.02	389.81	147.12	278.40

Figure 49: The history of solutions after the rearrangement illustrated in Figure 48.

was selected. After a selection a pop-up window with the text `Select the row to remove` is displayed (as shown on the bottom of Figure 48). After clicking on the label of the row to be removed, these two rows are exchanged as illustrated on Figure 49. In this example the third row was selected (by clicking on the label `C`) to be removed from the set of the displayed solutions.

5.4.4 Storing full solution

Figure 50: Confirmation of storing the solution.

The full information about the current solution can be saved by selecting the `Store` item from the `Solution` menu item of the main MCMA window. The file name for the full solution is selected automatically, and the dialog shown in Figure 50 confirms the actual filing of the current solution.

5.4.5 Resetting the problem analysis

In some situations it might be desirable to remove all the solutions from the history and start the problem analysis from scratch. This can be done by selecting the `Reset` item from the main ISAAP menu. In order to avoid undesired removal of the history, a confirmation of this action is requested by a dialog illustrated in Figure 51.

5.5 Files created by MCMA and ISAAP

There is a number of files created during the analysis of the problem. All files created by the MCMA have a root of a name equal to the root of the name of the file which

Figure 51: Dialog for a confirmation of the resetting of the problem analysis.

contains the core model. The contents of those files are documented in Section 4.7. Please do note that:

- contents of the files will be overwritten, if one starts the analysis of a new instance of the model. Therefore, before starting the analysis of a new instance one has to make sure that the files from the previous analysis that may be needed in future are moved to another place or are renamed.
- standard procedures for making back-ups should be implemented to avoid risk of loosing results from analysis.

5.6 Miscellaneous information on the AEZ example

The number of criteria used in this tutorial example has been limited to four in order to make it easier to publish the screen-dumps that illustrate the tutorial. A more complete analysis of the AEZ model (cf (Antoine et al., 1996)) uses seven criteria and the model has ten predefined variables that can easily be used as criteria for multi-criteria analysis. Therefore, we summarize here the convention used by the AEZ core model generator for the variables that can be used for definition of criteria. This information might be useful for those users who would like to make a more realistic analysis of the AEZ model. The names of the variables defining the criteria in the AEZ model 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 `FoodAv` (which is assigned the number 1) whereas the variable `V0000010` defines the criterion `MaxEro` (number 10 on the list of criteria, see (Antoine et al., 1996) for the list and interpretation of the all examined criteria).

5.7 Miscellaneous information on the Nitra example

For the Nitra case study (documented in (Makowski et al., 1995)) the definition of criteria can be done in the following way:

- In order to select only variables which can be used for environmental criteria type, enter the `cr` string as a mask name (cf the discussion related to Figure 13 on page 26) and click the mouse on the `OK` button. A window titled *Variables defining criteria* (similar to the window illustrated in Figure 14) will be displayed.
- Double click on the name `cr_0` to copy this name from the *Selected variables* list to the *Criteria defined by:* list. Do the same for the names `cr_1` and `cr_3`. You may

also click on the `Help` button to read a short information about this dialog. The variables `cr_0`, `cr_1` and `cr_3` represent three water quality constituents, namely DO (dissolved oxygen), CBOD and NH4 (ammonia), respectively.

- 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 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 with three more variables in the left window.
- Double click on the name `tot_inv`. Then the name `tot_inv` will be copied from the *Selected variables* list to the *Criteria defined by:* list. Do the same for the names `tot_omc` and `tot_tac`. The selected variables represent the total investment cost, the total OMC (operation and maintenance) cost and the total annual cost respectively. Click the mouse on the `OK` button.

Finally, please do note that all the selected criteria (except of DO, which should be maximized) for this case study should be minimized. Specification of the criteria type should be done by clicking on the corresponding radio buttons in the *Definition of criteria* dialog (see Figure 15). Using this dialog one can also define meaningful names for criteria.

6 Availability of software and documentation

The MCMA software is available from: `http://www.iiasa.ac.at/~marek/soft` free of charge for non-commercial research and educational purposes (please read carefully the license agreement which is available at the same URL on the Web). 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 7 for details). MCMA can easily be used for the analysis of any LP or MIP model provided that a corresponding *core model* is available in the LP-DIT format (Makowski, 1998) or in the MPS format.

This report serves as a documentation of ISAAP. Updated versions of this report will be made available, if need arises. Most of the reports related to the topics discussed in this paper are available on-line from the Web address given above. Hard copies can be ordered from the Publication Department of IIASA (orders can be placed also via WWW).

6.1 Installation of the software

All the executable files should be placed in any directory that is included in the `PATH`. The `mcma_hlp` directory should be placed in the same directory, where the `mcma` executable file will be located. This directory contains all the files that are needed for the on-line help. The `mcma` program will look for the help files in the `mcma_hlp` directory, therefore this directory should be located in the place specified above and must not be renamed.

The `mcma` software is distributed in the form of a `ZIP file` for the Unix machines and in the form of the self-extracting file for PCs. Therefore, the easiest way of installation of the `mcma` software is to *un-zip* the distributed file in one of the directories, which is included in the `PATH` (then the `mcma_hlp` directory will be created in a correct place).

6.2 Trouble-shooting

The authors of the software will try to do their best to help with using the software described in this paper. However, the authors may not be able to help in every case. In order to increase efficiency of developing and using the software, the users are kindly asked to first check, if the installation (or update) of the software followed the instructions provided in this paper.

In case of problems that can not be solved without help from the authors of the software, the user is advised to perform the following steps:

- Make a back-up of the working directory.
- Make a list of all files from the working directory. Such a list of files has to contain names, sizes and dates (including time) of all files located in the working directory.
- Prepare a detailed description of the problem.
- Write down exact specification of your hardware (which should include: type of the PC, amount of RAM, free disk space).

Please send to one of the authors an e-mail composed of the following elements (please follow the sequence specified below):

- Detailed description of the problem.
- The above specified list of files in your working directory.
- Specification of your hardware.
- Your e-mail address.
- Your full name, organization and postal address.

Depending on the type of problem, the user may be asked to ftp selected files.

Suggestions for improvements/extensions of the software are most welcome and will be dealt with as resources permit.

6.3 Updates of the software

The authors plan to continue further development of the software described in this paper. Therefore, it is likely that new versions of the MCMA package will be made available under the following URL:

- www.ia.pw.edu.pl/~janusz
- www.iiasa.ac.at/~marek

Users who would like to use the latest available version of the MCMA should check the above listed Web sites[17] and download updates of the software.

7 Conclusion

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

[17] It is planned to create mailing lists for users who will register their names on one of these Web servers.

- A DSS developed for the Regional Water Quality Management Problem, case study of the Nitra River Basin (Slovakia) documented in (Makowski et al., 1995; Makowski et al., 1996). This application is a result of the cooperation between the MDA and WAT Projects.
- Multi-Criteria Analysis in Optimizing Land Use for Sustainable Agricultural Development Planning described in (Antoine et al., 1996; Fischer et al., 1998). This application is a result of the cooperation of the LUC and MDA Projects with the FAO (Food and Agriculture Organization of the United Nations).
- Multi-Criteria Analysis of Urban Land-Use Planning, by Matsuhashi (1997).
- A number of engineering applications in mechanics, automatic control and ship navigation summarized by Wierzbicki and Granat (1997).

Other applications are planned in the near future.

The following extensions of ISAAP 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 ISAAP window.
- Analysis of history using an extension of the methodology described and applied for BIPLOT by Lewandowski and Granat (1991).
- Implementation of an interface to MCMA which will allow for its easy application to nonlinear problems.

The authors would appreciate comments and suggestions regarding functionality and robustness of ISAAP. 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.

8 Acknowledgment

The authors of the paper acknowledge and appreciate the contribution of Mr Grzegorz Wójcik of the Warsaw University of Technology, who has developed a tool for converting a single source documentation in the form of LaTeX (with additional styles) into three types of documents, namely a standard LaTeX document, the HTML files, and the restricted HTML files accepted by a portable *zHelp* utility. The tool generates also a dictionary which makes it possible to implement a context sensitive help in a C++ program.

References

Antoine, J., Fischer, G. and Makowski, M.: 1996, Multiple criteria analysis in optimizing land use for sustainable agricultural development planning, *Working Paper WP-96-06*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Antoine, J., Fischer, G. and Makowski, M.: 1997, Multiple criteria land use analysis, *Applied Mathematics and Computation* **83**(2–3), 195–215. available also as IIASA's RR-98-05.

Brooke, A., Kendrick, D. and Meeraus, A.: 1992, GAMS*, A User's Guide, release 2.25*, The Scientific Press, Redwood City.

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

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

Fischer, G., Granat, J. and Makowski, M.: 1998, AEZWIN an interactive multiple-criteria analysis tool for land resources appraisal, *Interim Report IR-98-051*, International Institute for Applied Systems Analysis, Laxenburg, Austria. Available on-line from `http://www.iiasa.ac.at/~marek/pubs`.

Fourer, R., Gay, D. and Kernighan, B.: 1996, *Using* AMPL *Plus*, Compass Modeling Solutions Inc., San Francisco.

Gardiner, L. and Steuer, R.: 1994a, Unified interactive multiple objective programming, *European Journal of Operational Research* **74**, 391–406.

Gardiner, L. and Steuer, R.: 1994b, Unified interactive multiple objective programming: An open architecture for accommodating new procedures, *Journal of Operational Research Society* **45**(12), 1456–1466.

Gondzio, J. and Makowski, M.: 1995, 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. Available on-line from `http://www.iiasa.ac.at/~marek/pubs`.

Granat, J. and Makowski, M.: 1995, SAP – Modular tool for specification and analysis of user preferences in multiple-criteria model analysis, *Working Paper WP-95-73*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

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

Inm: 1993, zApp *Interface Pack, Programmer's Guide & Reference.*

Inm: 1995, zApp*, The Portable C++ Application Framework, Programmer's Guide.*

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

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

Korhonen, P., Lewandowski, A. and Wallenius, J. (eds): 1991, *Multiple Criteria Decision Support*, Vol. 356 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York.

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

Lewandowski, A. and Wierzbicki, A.: 1989a, Decision support systems using reference point optimization, *in* A. Lewandowski and A. Wierzbicki (eds), *Aspiration Based Decision Support Systems: Theory, Software and Applications*, Vol. 331 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York.

Lewandowski, A. and Wierzbicki, A. (eds): 1989b, *Aspiration Based Decision Support Systems: Theory, Software and Applications*, Vol. 331 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York.

Makowski, M.: 1994a, Design and implementation of model-based decision support systems, *Working Paper WP-94-86*, International Institute for Applied Systems Analysis, Laxenburg, Austria. Available on-line from `http://www.iiasa.ac.at/~marek/pubs/`.

Makowski, M.: 1994b, 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. Available on-line from `http://www.iiasa.ac.at/~marek/pubs/`.

Makowski, M.: 1996, Methodology and modular tools for aspiration-led analysis of LP models, *in* J. Doležal and J. Fidler (eds), *System Modelling and Optimization*, Chapman & Hall, London, New York, Tokyo, pp. 371–378.

Makowski, M.: 1998, LP-DIT++, C++ class library for Data Interchange Tool for Linear Programming Problems, (version 2.06), *Interim Report IR-98-xxx*, International Institute for Applied Systems Analysis, Laxenburg, Austria. (to appear).

Makowski, M., Somlyódy, L. and Watkins, D.: 1995, 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. Available on-line from `http://www.iiasa.ac.at/~marek/pubs`.

Makowski, M., Somlyódy, L. and Watkins, D.: 1996, Multiple criteria analysis for water quality management in the Nitra basin, *Water Resources Bulletin* **32**(5), 937–951.

Matsuhashi, K.: 1997, Application of multi-criteria analysis to urban land-use planning, *Interim Report IR-97-091*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

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

Ogryczak, W. and Lahoda, S.: 1992, Aspiration/reservation-based decision support — a step beyond goal programming, *Journal of Multi-Criteria Decision Analysis* **1**(2), 101–117.

Ogryczak, W. and Zorychta, K.: 1996, Modular optimizer for mixed integer programming, MOMIP version 2.3, *Working Paper WP-96-106*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

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

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

Stewart, T.: 1992, A critical survey on the status of multiple criteria decision making theory and practice, *OMEGA, International Journal of Management Science* **20**(5/6), 569–586.

Wessels, J. and Wierzbicki, A. (eds): 1993, *User-Oriented Methodology and Techniques of Decision Analysis and Support*, Vol. 397 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York.

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

Wierzbicki, A.: 1980, The use of reference objectives in multiobjective optimization, *in* G. Fandel and T. Gal (eds), *Multiple Criteria Decision Making, Theory and Applications*, Vol. 177 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York, pp. 468–486.

Wierzbicki, A.: 1982, A mathematical basis for satisficing decision making, *Mathematical Modelling* **3**(5), 391–405.

Wierzbicki, A.: 1986, On the completeness and constructiveness of parametric characterizations to vector optimization problems, *OR Spektrum* **8**, 73–87.

Wierzbicki, A.: 1992, Multi-objective modeling and simulation for decision support, *Working Paper WP-92-80*, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Wierzbicki, A. and Granat, J.: 1997, Multi-objective modeling for engineering applications in decision support, *in* G. Fandel and T. Gal (eds), *Multiple Criteria Decision Making*, Vol. 448 of *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, New York, pp. 529–540.

Wierzbicki, A., Makowski, M. and Wessels, J. (eds): 1999, *Model-Based Decision Support Methodology with Environmental Applications*, Kluwer Academic Publishers, Dordrecht, Boston, London. (to be published).

Zimmermann, H.: 1978, Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Systems* **1**, 45–55.

Zimmermann, H.: 1985, *Fuzzy Set Theory – and Its Applications*, Kluwer Academic Publishers, Boston, Dordrecht, Lancaster.

Zimmermann, H.: 1987, *Fuzzy Sets, Decision Making, and Expert Systems*, Kluwer Academic Publishers, Boston, Dordrecht, Lancaster.

Index