Raport Badawczy

Research Report

RB/35/2017

Structured modeling infrastructure for multiple-criteria decision aid

M. Makowski, H. Ren,
S. C. Parkinson

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences

POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Zakładu zgłaszający pracę:
Prof. dr hab. inż. Zbigniew Nahorski

Warszawa 2017

Structured modeling infrastructure for multiple-criteria decision aid

Marek Makowski[a,b,*], Hongtao Ren[c], Simon C. Parkinson[a]

[a] *Energy Program, International Institute for Applied Systems Analysis, Laxenburg, Austria*
[b] *Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland*
[c] *Business School, East China University of Science and Technology, Shanghai, China*

Abstract

Rational decision-making in complex situations requires well structured model-based support. The paper presents methodology of modular modeling environment providing such support. The environment has been designed and implemented for effectively handling large-scale complex models, and for enabling model analysis also by users without modeling skills. The features of the developed system were motivated and tested by diversified case-studies in different fields. In particular, the underlying models are characterized by criteria with possibly huge ranges of criteria values, and by long computation time of the resulting optimization tasks. Such model characteristics demand advanced methods for the model analysis. The paper presents the effective implementation of the modular, web-based multiple-criteria model analysis tool that can be used with different model-development environments without any modifications of the latter. Moreover, the paper presents the energy-water-climate nexus problem analyzed with the developed tool.

Keywords: decision-making support, multicriteria model analysis, structured modeling, modeling systems and languages, large-scale complex models, energy-water-climate nexus, model management.

1. Introduction

The presented work as well as countless other applied research activities are motivated by the needs of rational support for solving of complex decision-making problems. Such decisions are of different types, including not only various policy or management problems, but also choices made in diverse areas of e.g., industry, research, medical treatment, and infrastructure planning. The background knowledge on decision-making and support is well researched since long ago in all related (and often overlapping) disciplines, e.g., management [1, 2, 3, 4, 5, 6], psychology [7, 8, 9, 10, 11, 12], as well as mathematics and operational research [13, 14, 15, 16, 17, 18, 19, 20, 21]. Yet, the large gap between the knowledge and its effective implementation (pointed out already decades ago, see

*Corresponding author. *Email address:* marek@iiasa.ac.at

e.g., [22, 23, 24, 25, 26, 27]) still exists despite of the ever growing experience. The research described in this paper aims at narrowing this gap.

As a basis for presenting the objectives and scope of the work we summarize several commonly known facts:

1. Decision problems are more and more complex; rational dealing with them usually requires science-based support. Science develops knowledge often using mathematical models in two stages: first, for integrating and representing relevant knowledge; second, for creating, through integrated model analysis, knowledge aimed at problem understanding and aiding rational decision-making.
2. Problem understanding requires reliable and transparent representation of relations between decisions and consequences of their implementation; the latter are measured by several (or many) criteria. Effective decision-making support has to guarantee the Decision-Maker (DM[1]) sovereignty in exploring all solutions through transparent specification of preferences. DMs are not only persons making policy or management decisions, but all who are involved in decision-making processes, e.g., advisors, experts, as well as e.g., scientists, engineers, physicians, who make decisions in their work.
3. Model-based decision-making support requires properly structured modeling environment composed of humans, workflows, processes, and computing infrastructure including software tools. Such environments require substantial resources therefore reusing (or adapting) modular software is a must.
4. Multiple-Criteria Analysis (MCA) methods and tools are key components of integrated analysis of any non-trivial decision-making problem. Diverse MCA methods and tools have been developed to address needs of different types of problems. Proper methods, tools and processes greatly enhance decision support quality while improper support labels as optimal solutions often far away from rationality.
5. Diverse characteristics of the problems and the associated decision-making processes call for diverse modeling approaches; however, methods and tools developed for a specific problem often can be reused or adapted for other problems, even of very different nature.

The paper deals with development and analysis of models representing corresponding decision-making problems. The class of such problems and the users (problem owners and analysts) is implicitly defined by the following key attributes of the model and its analysis:

1. specification for user preferences in the model analysis should enable analysis also by users without modeling knowledge and skills;
2. large ranges (several orders of magnitude) of criteria values;
3. large-scale models requiring long (several hours, or even days) wall-clock time for computations of optimization tasks;

[1] Abbreviations frequently used in this article: A/R - aspiration/reservation criteria values; MCA - multiple-criteria analysis (in general); MCAA - multiple-criteria analysis of a given set of discrete alternatives; MCMA - multiple-criteria model analysis (in general; also the name of the software tool described in this paper);

4. background model management and runs of optimization tasks should be transparent for users not interested in software technology;
5. diverse modeling environments used for model development.

The leading objectives of the presented work are threefold. First, development of methodology and corresponding software tools for supporting decision-making for problems of the above defined class. Second, integration of these methods and tools into a modeling environment suitable for Multiple-Criteria Model Analysis (MCMA). Third, application of such environments to real-life complex problems.

The above objectives have been met through long-term activities interlinking development of methods and tools with their applications to complex problems. Therefore, the actual objectives content have been periodically extended by challenges incoming from more and more demanding applications. The above characteristics of the class of problems determine the requisites for the modeling technology, i.e., the methods and tools, in particular for MCMA. Section 4.3 summarizes how the corresponding requirements are met by the implemented modeling environment.

The paper reports also the energy-climate-water nexus model and its analysis, as well as the lessons from this case-study. The lessons are of two types: first, pertinent to the modelled problem; second, contributing to the MCMA experience and thus useful for further developments of MCA methods and tools.

The remaining part of the paper is organized as follows. Concepts and notations are presented in Section 2; Section 3 follows with the theory and methods underlying the implementation of the modeling environment presented in Section 4. The energy-climate-water nexus model and its analysis are discussed in Section 5. The paper concludes by Section 6.

2. Concepts and notations

The main purpose of model-based problem-solving (aka decision-making) support is to analyze a given problem in terms of relations between decisions and consequences of their implementation in order to aid finding decisions that fit best the preferences of actual DMs. Such problems are typically composed of an underlying system (e.g., interlinked energy, water, climate subsystems), and includes decisions (e.g., choice of technologies, capacities) aimed at controlling the system state or behavior. This topic has been researched since long ago and countless publications report diverse methodologies and experience.

Model-based decision support is composed of two main stages. (1) development of a model representing the relations between decisions and consequences, and (2) processes and tools for analyzing these relations. This paper focuses on the second stage, therefore we restrict the discussion of the first stage to issues necessary to understanding these elements of the model development that determine the analysis quality.

Moreover, we designed and implemented modular architecture in which model development is encapsulated, i.e., it is separated from the MCMA. Therefore, modular MCMA can easily be linked into the corresponding modeling process and thus is available for diverse methods and tools for the model development.

2.1. Structured modeling processes

Mathematical modeling is a wide and diversified research area, discussion of which is clearly beyond the scope of this paper. Readers interested in modeling methodology and tools may want to explore rich literature, e.g., [20]. Here we outline the modeling process elements directly related to main scope of the paper.

A mathematical model (further on: model) is an abstract representations of the corresponding piece of reality developed for a specific purpose. Diverse modeling methods and tools have been developed for different needs and preferences. We have aimed at providing MCMA capable to interface different modeling tools. In order to present the underlying architecture we exploit the *Structured Modeling* (SM) paradigm developed by Geoffrion [28]. The SM provides a proven methodological background for effective modeling processes, including consistency of model specification, data used for model parameters, model analyses, and interpretation of results.

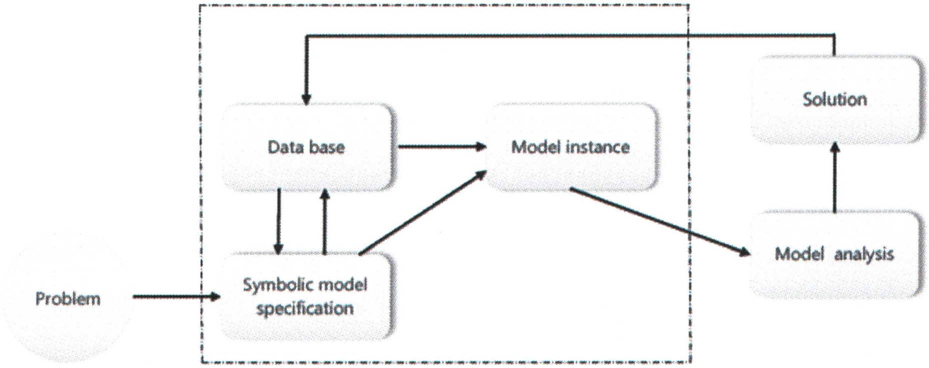

Figure 1: Basic elements of structured modeling.

Figure 1 illustrates the main elements of well structured modeling process summarized below:

- Modeling process starts with a thorough understanding of the decision problem and exploration how to represent the pertaining knowledge by a Symbolic Model Specification (SMS), which is an abstract representation of the corresponding problem by three types of entities: variables, relations between variables, and parameters used in specification of the relations. Section 2.2 summarizes the SMS, in particular the concepts applied to the modular architecture described in this paper.
- A Data-Base (DB) used for management of data, in particular model parameters and results of model analysis. DB is a commonly known and widely used concept therefore need not to be discussed here.
- Model instance composed of selected versions of the SMS and of the corresponding data

defining parameters. The model instance concept comes from the SM methodology; it greatly helps structuring the modeling process, in particular in interfacing modular tools for developing SMS, DB, and diverse methods and tools for model analysis.

- Model analysis.

2.2. Symbolic model specification

We present here a specific view on modeling, simplified for, and tuned to the paper purpose. We start with introducing compound entities, the concept used through the paper, and follow with summarizing two views on model specification: from mathematical programming and decision-making perspectives, respectively.

2.2.1. Compound entities

For modeling efficiency and consistency, entities of all three types (variables, relations, parameters) shall effectively share common attributes, and be organized into structures useful for model development and analysis. In terms of Object Oriented Programming (OOP) diverse entities can be derived from a common abstract class; details of this approach are provided e.g., in [29], and are not discussed here. Modelers using diverse modeling systems may not be aware of this methodological background. However, proper structuring variables, parameters and relations is essential for any modeling system. Such a structuring can be achieved in two ways: (1) by structuring single entities into compound entities, and (2) by organizing entities into compound entities according to diverse needs, e.g., their roles. We illustrate these concepts by simple examples.

Single entities are actually used in mathematical programming tasks, while compound entities are sets of the corresponding single entities. In order to illustrate this concept let us consider variables representing air pollution concentration defined for combinations of time-periods, regions, and pollution types. Such a set of single variables can be structured into a compound variable, e.g., **conc** defined as:

$$\mathbf{conc} = \{conc_{t,r,p}\}, \qquad t \in T,\ r \in R,\ p \in P_r, \tag{1}$$

where T and R are sets of time-periods and regions, respectively; P_r is a set of pollution types considered in r-th region. A given set of indices and the corresponding sets are referred to as indexing structure.

Compound entities can be further organized into containers (higher-level entities), e.g., $\mathbf{u}$ composed of all variables representing decisions, see Section 2.2.3. Moreover, it also might be useful to optionally define indexed subsets of compound variables, e.g., $\mathbf{u}_t \in \mathbf{u}$ composed of all variables representing decisions make in t-th period.

Compounding is a simple but powerful concept when properly used; in particular, instantiation of sets of single variables and relations from the corresponding compound entities through indexing structure has several advantages: it provides flexibility (e.g., through modifications of indexing sets), it supports consistency between parameters, variables, and relations, it helps in generating model documentation and diverse views on model results; finally, it eases interfacing modular modeling tools.

2.2.2. Mathematical programming view on SMS

A model represents the corresponding problem by two types of entities: variables and relations between them. Many problems, including the case study in Section 5, are described by linear, often dynamic and spatial, models. A standard mathematical programming formulation of such models takes the form:

$$\underline{\mathbf{x}} \leq \mathbf{A} \cdot \mathbf{x} \leq \bar{\mathbf{x}} \tag{2}$$

where vector $\mathbf{x}$ is composed of all model variables, and the matrix $\mathbf{A}$, as well as vectors $\underline{\mathbf{x}}$ and $\bar{\mathbf{x}}$ are the model parameters.

The set X_f of feasible model solutions $\check{\mathbf{x}}$ is defined by:

$$X_f = \{\check{\mathbf{x}} : \underline{\mathbf{x}} \leq \mathbf{A} \cdot \check{\mathbf{x}} \leq \bar{\mathbf{x}}\}. \tag{3}$$

If model (2) represents only logical and physical relations between variables defining a decision problem, then the set X_f is composed of many (typically, infinite number of) elements. Feasible solutions are defined implicitly by (2); therefore a completeness of set X_f can only be assured by the model verification. An empty set X_f indicates errors in either the model specification (2) or the data defining the model parameters. However, such errors often cause other, more difficult to diagnose, problems; namely an incomplete (although non-empty) set X_f, i.e., a set that does not include all actually feasible solutions.

2.2.3. The user view on SMS

Complex models include many variables, typically hundreds of thousands or even millions. Therefore, specific and analysis analysis of the general model formulation (2) is impractical for the decision-making support needs. Such needs can be addressed by structuring the model through four types of compound variables, each representing a particular model role.

- $\mathbf{s}$ – The system state; e.g., temporal and spatial distribution of capacities, flows, various costs.
- $\mathbf{u}$ – Decisions (controls) considered for changing the system state; e.g., technologies for extending or phasing-out electricity supply capacities or water management.
- $\mathbf{y}$ – Outcomes measuring the system state, in particular the consequences of decision's implementation; e.g., total costs, CO_2 emission, ground-water extraction, uncovered fractions of electricity and water demands.
- $\mathbf{z}$ – Auxiliary, i.e., all other variables, i.e., variables not interesting for model users but useful for rational development of the SMS. In large models such variables constitute a vast majority of variables; this is also a reason to distinguish $\mathbf{s}$, $\mathbf{u}$, and $\mathbf{y}$ variables.

We focus on analysis of decisions aimed at improving the state of the system under consideration. The following specification, typical in control engineering, illustrates well such relations. In the example below we use only the t subscript to show the system dynamics; other indices (e.g., representing spatial resolution, or technologies) are omitted.

$$\mathbf{s}_{t+1} = B_t \mathbf{s}_t + C_t \mathbf{u}_t, \qquad t \in T, \qquad \mathbf{s} \in S_0, \qquad \mathbf{u} \in U_0, \tag{4}$$

where $\mathbf{s}$, $\mathbf{u}$ are vectors of the system state and decision variables, respectively; $\mathbf{s}_0$ is a given initial state of the system, matrices B_t, C_t are parameters, T is a set of periods, S_0 and U_0 are sets of feasible states and decisions, respectively. Sets S_0 and U_0 are typically defined implicitly, i.e., by additional relations augmenting (4). Note, that a one-step decision-making situation can be represented by model (4) having one-element set T. On the other hand, the template equation (4) is often generalized; in particular, for handling typical elements of dynamic process, such as delays, accumulations, and deteriorations of e.g., investments, equipment. These and many other issues of model specification are beyond the scope of this paper; discussion on related topics can be found e.g., in [30].

A complex system state is characterized by many $\mathbf{s}$ variables; therefore, a small set of outcome variables $\mathbf{y}$ is defined for the first stage analysis augmented by more detailed analysis in the state variables $\mathbf{s}$ sub-spaces. The latter analysis often leads to modifications of the $\mathbf{y}$ specification. Outcome variables are named differently in various application fields, e.g. outcomes, criteria, objectives, goals, indicators, metrics, (performance) indices, attributes. In this paper we call such variables outcomes. Formally, outcomes are implicitly defined by decisions $\mathbf{u}$ and system states $\mathbf{s}$:

$$\mathbf{y} = \mathbf{F}(\mathbf{s}, \mathbf{u}), \qquad \mathbf{y} \in Y_f \subset R^m, \tag{5}$$

where m is the number of outcome variables, $\mathbf{F}(\cdot)$ denotes a set of relations defining $\mathbf{y}$, and Y_f is the set of feasible outcomes; note that, by definitions, $Y_f \subset X_f$. Relations (4) and (5) are amalgamated into (2); therefore, they are rarely considered explicitly.

2.2.4. Compound and outcome variables summary

Compound entities and outcome variables are the two concepts not yet widely used but key for the approach described in this paper. Therefore, we summarize here two corresponding issues.

First, depending on the context, one considers either all model variables $\mathbf{x}$ or its split into components according to the role diverse compound variables represent:

$$\mathbf{x} = \{\mathbf{y}, \mathbf{u}, \mathbf{s}, \mathbf{z}\}. \tag{6}$$

Accordingly, we focus in Sections 3 and 4 on either all or outcome variables. Comments on the state and decision variables are included in Section 5.

Second, we summarize the role and desired features of outcome variables, and the corresponding guidelines for their specification.

- One should define in SMS as many outcome variables as helpful for evaluating different aspects of the model solution. Such definitions are very easy during the SMS development and do not increase computational requirements. A surplus (compared to typically small number of outcomes initially considered for criteria) of outcome variables increase flexibility and efficiency of model instance analysis; moreover, outcome variables are also helpful in model verification.

- Note, that (5) defines $\mathbf{y}$, therefore usually neither lower nor upper bounds for $\mathbf{y}$ are appropriate for representing logical and physical relations. In particular, it is important to refrain from representing preferences by constraining $\mathbf{y}$ values in SMS; such bounds typically cause problems in the model analysis; Section 3.1 provides more comments on this issue.
- Outcome variables are actually used in traditional (single-criterion) optimization approaches to model analysis, see Section 3.1.
- Subsets of outcome variables serve as criteria in MCA of a model instance, see Sections 2.3 and 3.1.

2.3. MCA preliminaries

2.3.1. Outcomes and criteria

Before summarizing the key concepts of MCA we comment on the relations between outcomes $\mathbf{y}$ defined in SMS and the criteria $\mathbf{q}$ interactively defined during the MCMA:

$$\mathbf{q} \in \mathbf{y}, \qquad \mathbf{q} \in Q \subset R^n, \tag{7}$$

where n is the number of selected criteria. Thus, $n \leq m$, and $Q \subset Y$. In other words, each criterion is defined by the corresponding outcome variable. This approach might be surprising; therefore, we provide two key arguments. First, the approach enables separation of the SMS development from the MCMA, and thus provides a very effective linkage between them by outcomes $\mathbf{y}$. Second, it provides MCMA with flexibility of selection (without model instance modifications) diverse criteria sets that fit different user preferences.

2.3.2. Preference structure

We briefly summarize basic concepts of preference used in this paper. A comprehensive discussion of preference models can be found e.g., in [14, 16, 31, 32]. The simplest preference model assumes that when two elements are being compared only two situations can be distinguished: preference of one element to the other (relation $\succ$), or indifference of one element to the other (relation $\sim$). In this paper we assume even simpler preference model that is still adequate for this paper presentation.

Assume, only for simplifying presentation of this preference model, all criteria to be minimized. Consider two vectors $\mathbf{q}^1$, $\mathbf{q}^2$ of criteria values; then:

$$\mathbf{q}^1 \succ \mathbf{q}^2 \qquad \Longleftrightarrow \qquad (\mathbf{q}^1 \leq \mathbf{q}^2 \text{ and } \mathbf{q}^1 \neq \mathbf{q}^2) \tag{8}$$

Moreover, if neither $\mathbf{q}^1 \succ \mathbf{q}^2$ nor $\mathbf{q}^2 \succ \mathbf{q}^1$ then $\mathbf{q}^1 \sim \mathbf{q}^2$. Indifference means that the two entities are not objectively comparable.

Further on, following the common practice, definition (8) is used in terms of dominance, i.e., $\mathbf{q}^1$ dominates $\mathbf{q}^2$ if, and only if $\mathbf{q}^1 \succ \mathbf{q}^2$. Moreover, we use such dominance concept also to other entities of MCMA.

In the plain language, $\mathbf{q}^1$ is preferred over $\mathbf{q}^2$, if all $\mathbf{q}^1$ values are at least as good (smaller or equal for minimized criteria) as the corresponding values of $\mathbf{q}^2$, and at least one criterion value is strictly better (i.e., smaller). We use the dominance (preference) concept because it simplifies the presentation, in particular enables uniform treatment of minimized and maximized criteria, as well as combinations of these criteria types; the latter very often occurring in practice.

2.3.3. Pareto efficiency

Noting that $\mathbf{q} \in Q \subset Y \subset X$, also here we simplify the presentation. Namely, we refer to $\mathbf{q}$ as a vector composed of all criteria values that belong to a solution $\mathbf{x}$. For short, we also often use the term *solution* $\mathbf{q}$ for vector of criteria values belonging to the corresponding solution $\mathbf{x}$. Analogously, $\check{\mathbf{q}}$ and $\hat{\mathbf{q}}$, denote attainable (feasible) and optimal solution in criteria space Q, respectively. Finally, the set of all attainable $\check{\mathbf{q}}$ is denoted by Q_f.

A solution $\hat{\mathbf{q}}$ is called Pareto-efficient, if there is no other feasible solution that dominates it. Other attributes are often used instead of *Pareto-efficient*, e.g., non-dominated, Pareto-optimal, or simply Pareto. For the sake of brevity we don't deal here with more advanced concepts, e.g., properly efficient solutions; these are discussed e.g., in [20].

2.3.4. Pareto set

Set of all Pareto solutions is denoted by Q_P and is defined by:

$$Q_P = \{\hat{\mathbf{q}} \,:\, (\nexists\, \check{\mathbf{q}} \,:\, \check{\mathbf{q}} \succ \hat{\mathbf{q}}\,)\} \tag{9}$$

Pareto set and its characteristics are key concepts of MCA. Therefore, we summarize the basic corresponding attributes.

Figure 2 illustrates Q_P for an example of two minimized criteria. For the discussion sake, let q_1 represent cost, and q_2 the CO_2 emission. The Q_P set is marked by the thick line; extreme points of Q_P (marked by E and D) correspond to the best solutions in terms of q_1 and q_2, respectively; e.g., point E shows the minimum value of cost and the corresponding worst (maximum within Q_P) value of CO_2. The broken top Q_P segment (and the corresponding break of the q_2 axis) illustrate the situation in which huge compromise in one criterion is needed for relatively small improvement of another criterion, i.e., a minimum cost solution would result in huge emission, while a relatively small increase of cost results in dramatic emission decrease. Section 5 presents real-life examples of solutions close to both extremes (close to utopia and nadir values, respectively) of the cost criterion.

The point U is defined by the best values of both criteria, q_1^U, q_2^U, respectively; this point is called *Utopia* because it is not attainable. The point Nadir point denoted by N is defined by the worst (within Q_P) values of both criteria, q_1^N, q_2^N, respectively. In the shown example it is attainable, but this is not always the case.

Selected Pareto solutions are marked by letters C, K, P, L, and M. Solution D is expensive but clean, solution E is cheap but has high emission; solutions located on the

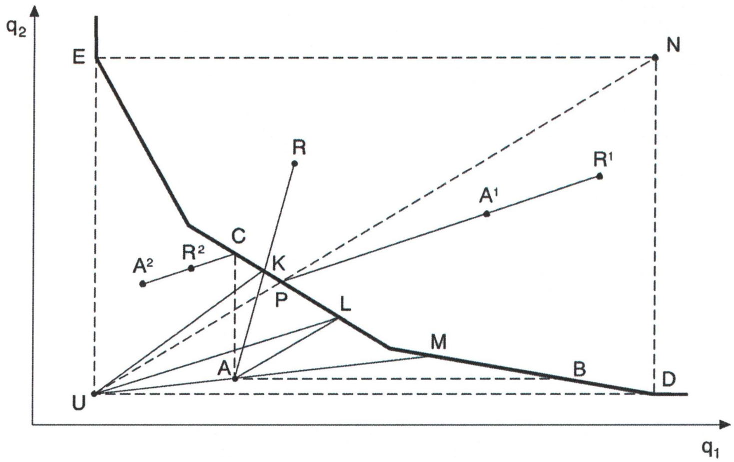

Figure 2: Pareto set, utopia, nadir, aspiration, and reservation points.

Pareto frontier between these two extreme solutions match different trade-offs between costs and the corresponding emission; solutions B and M are substantially cheaper than D and for both the corresponding worsening of emission is substantially smaller than for yet cheaper solutions L, K, and C. More comments on these solutions, as well as on the points marked by letters A and R, are available in Section 3.3.

Analysis of trade-offs between the criteria values, and finding the solution having a preferred trade-off is easy for two-criteria problems; actually for such problems MCA is not recommended, see Section 3.2. However, for problems with more criteria MCA recommended for exploring Q_P representation that help the users for find solutions fitting their preferences. The concepts of dominance, Pareto set, as well as Utopia and Nadir are very helpful in such exploration.

3. Theory and methods

3.1. Model instance analysis

A selected model instance (composed of given versions of SMS and of data defining parameters) implicitly defines relations between decisions and outcomes. The main purpose of the instance analysis is to find decisions that result in outcomes possibly best fitting the user preferences. As shown by (6) both decisions and outcomes are components of a solution $\check{\mathbf{x}}$ of the corresponding mathematical programming problem (2). In other words, one attempts to generate such a solution of (2) that meets the user preferences (represented by outcomes $\mathbf{y}$) and defines decisions $\mathbf{u}$ leading to the desired outcomes.

However, an a priori definition of desired outcomes that are attainable is impossible for complex decision problems. This widely known observation had led to commonly

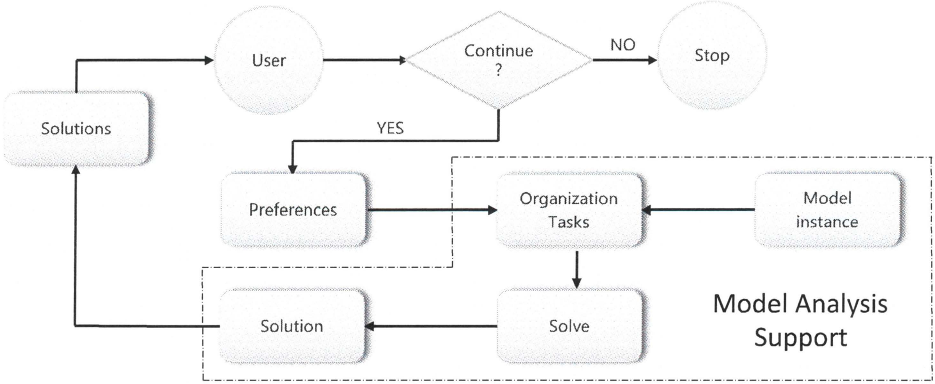

Figure 3: Iterative process of model instance analysis for decision support.

applied iterative process of model instance analysis, shown in Fig. 3. The process is composed of the following steps controlled by the user:

1. Analyze previously obtained solutions and decide either to break the analysis (e.g., because one of solutions is satisfactory or no more analysis progress is recognized) or decide to continue analysis, and select a solution to serve as the base for finding another one with more preferred outcomes.
2. Define preferences for improvement of outcomes $\mathbf{y}$.
3. Supporting software or staff represents the preferences in terms of mathematical programming, i.e., together with the model instance define the corresponding optimization task.
4. The optimization task is passed to a suitable solver; solution of such a task, traditionally denoted $\hat{\mathbf{x}}$, is added to the set of optimal solutions.
5. The process continues with step 1.

Formally, the above process can be defined as:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x} \in X_f} \mathcal{P}(\mathbf{y}), \tag{10}$$

where $\mathcal{P}(\cdot)$ represents the user preferences. The preferences can, of course, be also specified in terms of all model variables, as well as be minimized instead of maximized. Simple transformations of (10) can handle such, and several other, alternative formulations of optimization problems.

Summing-up: optimization serves as a tool for selecting from the set of all feasible solutions X_f one solution that fits best the user preferences $\mathcal{P}(\cdot)$. The users modify specification of $\mathcal{P}(\cdot)$ while learning diverse attainable outcomes during the iterative process outlined above. Effectiveness of such a process critically depends on a clear distinction of the two submodels of the optimization model:

- Objective, in which a complete X_f is defined by (2) representing in the selected model instance only relevant physical and logical relations between decisions and outcomes, i.e., core model (2) does not include constraints representing preferences.

- Subjective, representing preferences $\mathcal{P}(\cdot)$ that are iteratively modified by the user.

An aggregation of both submodels enables solution of (10) and thus provides a solution optimizing the given specification of $\mathcal{P}(\cdot)$. Such solutions, corresponding to diverse specifications of $\mathcal{P}(\cdot)$, implicitly define set X_a of admissible (i.e., acceptable in terms of the user preferences) solutions. Obviously:

$$X_a \subset X_f. \tag{11}$$

Distinguishing feasible (objectively possible) and admissible (acceptable for users) solutions is helpful from both methodology and implementation perspectives. Acceptability depends on preferences that usually differ amongst the users; moreover, often a particular user changes, sometime dramatically, preferences during the model analysis. Feasibility is independent of preferences; it is determined by the SMS and selection of data defining the model parameters, and should be verified before model analysis start.

The formulation (10) covers all optimization-based approaches to model analysis. Before discussing in Section 3.3 the MCMA methods we briefly characterize more traditional approaches.

3.2. Traditional approaches

Most approaches amalgamate preferences into model (2) and use a selected model variable or a relation as optimization criterion, traditionally called Goal Function (GF).[2] A similar approach consists of using as GF a so-called utility function, which represent a priori specified preferences. Usually, application of a GF is augmented by specification of bounds on selected outcome variables. Such approaches have widely recognized shortcomings; we refrain from commenting them here, as they are discussed in many publications, see e.g., [23, 33, 21, 34].

However, the traditional methods, known as *Goal Programming* (originating from [35]) provided a basis for a family of methods referred to as Reference-Point (RFP). These methods are based on the two-step approach:

- define a reference (aspiration, goal) point composed of the desired values of all criteria; one of the first RFP methods [36] used the displaced utopia point as the aspiration.
- find a Pareto solution that is (in a sense) closest to this point.

However, such two-stage procedure causes two types of problems in effective representation of user preferences:

- Selection of a measure for the distance between the RFP and the Pareto set; it is hardly possible for users to effective control the distance definition.
- The method provides dominated solutions, if the specified goal is attainable.

Figure 2 illustrates both problems. Different distance definitions make any Pareto solution between points B and C to be closest to the goal defined by point A. The goal

[2]In the traditional formulation of Linear Programming (LP), the first neutral row, often called cost function, serves as GF.

defined by point A^1 will be reached for any distance definition, i.e., a non-Pareto solution will be found.

The experience with RFP methods led to the development of theory and software for aspiration-based decision support methods, summarized in [18]; relations of these methods to the Goal Programming are summarized in [37]. Another stream of the developments in this field is outlined in [38]. A comprehensive discussion of the theoretical background of the reference point methodology, tools for their implementation as well as a detailed presentation of several applications can be found in [20].

3.3. MCMA theory

The MCMA method and implementation described in this paper has actually been pioneered by the ISAAP [39]. The ISAAP methodology has been enhanced, and the implementation reworked. Both elements were motivated by the challenges of the class of demanding problems characterized in Section 1 and by the qualitative improvement of the software technology capacity since the ISAAP release in 1990s. The MCMA methodology is presented next, and its implementation in Section 4.

The applied MCA Aspiration-Reservation methodology, for short called here the AR approach, is an extension of the aspiration-led MCA outlined above. It is composed of two interlined elements:

- Criterion Achievement Function (CAF); its role is to map the user preferences and the criteria values into a comparable scale. CAF is discussed in Section 3.3.1.
- Scalarizing Function (SF) aggregating CAFs into a GF used in parametrized single-criterion optimization problem that provides a Pareto-solution fitting best the user preferences. Section 3.3.2 presents the SF details.

Further on, n denotes number of criteria, and index $i \in \{1, \dots, n\}$ is used for entities of i-th criterion; e.g., $\mathbf{q} = \{q_1, \dots, q_i, \dots, q_n\}$.

3.3.1. Criterion Achievement Function

A Criterion Achievement Function (CAF) is defined for each criterion independently; therefore, for the brevity, we skip in this Section the criterion index. Thus, a selected criterion value (i.e., an element of $\mathbf{q}$) is here denoted q; similarly, q^u, q^a, q^r, q^n denote the utopia, aspiration, reservation, and nadir values, respectively. Index i is used only when needed, i.e., q_i^u denotes the utopia value of i-th criterion, utopia values of all criteria are denoted by $\mathbf{q}^u$.

CAFs have been proposed long ago, see e.g., [40]. The functions had different names and specifications, but their role was the same; namely, to map criteria values, specified in diverse measurement units and/or scales, into a common measure of criteria performance.

MCMA uses the CAFs that non only conform to the necessary conditions (specified e.g, in [14, 31]) but also enable handling of criteria taking values from even huge ranges. We denote CAF for i-th criterion by $caf_i(\cdot)$. The corresponding CAF properties are presented below.

The shapes of CAFs are illustrated in Figure 4 for maximized and minimized criteria, respectively. Strict monotonicity of $caf_i(\cdot)$ is the main necessary property. We adopt

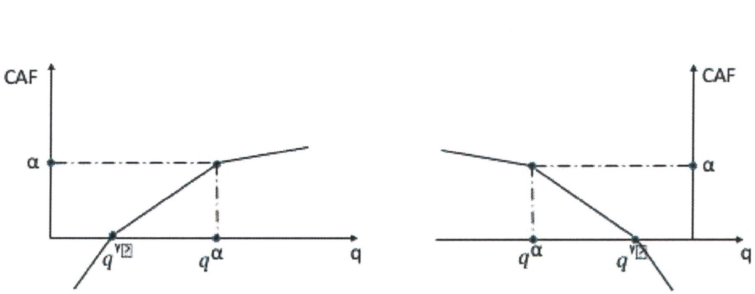

Figure 4: Criterion achievement functions for: maximized (left), and minimized (right) criteria, respectively.

the CAF's interpretation in terms of achievements, i.e., *the more the better*. Therefore, CAFs are strictly increasing/decreasing for maximized/minimized criteria, respectively. Finally, $caf_i(q_i)$ needs to be defined for all values between q_i^n and q_i^u.

Following the common practice, CAFs are specified as Piece-Wise Linear (PWL) functions. Such definition enables CAFs representation as a part of a Linear Programming (LP) problem. Thus, optimization MCMA tasks are of the same type as the model instance; therefore, the same solver can be used for both single- and multiple-criteria analysis.

CAFs are parametrized by the user preferences; namely, by the two desired criterion values:

- q_i^a - aspiration (goal), i.e., the criterion value the user wants to achieve, and
- q_i^r - reservation, the worst criterion value the user considers acceptable.

The UI of the MCMA enforces only the obviously justified condition:

$$q_i^u \succ q_i^a \succ q_i^r \succ q_i^n. \tag{12}$$

In other words, the user can specify any values of aspiration and reservation provided they conform to (12). In particular, it does not matter whether or not the specified values are attainable. Various combinations of aspiration and reservation (marked by A and R, respectively) are shown in Figure 2. For example, the pair (A^2, R^2) defines preferences typical for initial analysis stage, when often non-attainable reservation values are specified. The values shown by (A, R) are typical for users already having good feeling of realistic expectations. Preferences defined by (A^1, R^1) are rare, but still occur for problems with many criteria, especially, when in a previous iteration a non-attainable reservation was specified.

Each $caf_i(\cdot)$ is defined by a three-segments PWL function, parametrized by the user preferences represented by q^u and q^r. The middle CAF segment is defined by these values, and by the corresponding function values set by MCMA:

$$caf_i(sc_i(q_i^a)) = \alpha, \qquad caf_i(sc_i(q_i^r)) = 0, \tag{13}$$

where sc_i is the scaling transformation described below. The remaining segments are defined by the corresponding end-point of the middle segment and by the slope determined by the MCMA implementation.

In typical CAF implementations no criteria-value scaling is used, α is set equal to 1, and slopes of outside segments are defined either by the utopia and nadir points, or by multipliers of the middle segment slope. Such approaches work well for well-scaled problems, i.e., ranges of criteria values that do not cause numerical problems for CAFs defined in traditional ways. However, these approaches cause problems for model instances characterized by huge (or very small) ranges of criteria values.

Detailed specification of the scaling transformation is beyond the paper scope. Therefore, we only summarize the main features of the resulting CAF:

- Auxiliary variables and linear transformations are defined in such a way that

$$|sc_i(q_i^a) - sc_i(q_i^r)| = \alpha, \qquad sc_i(q_i^r) = 0. \tag{14}$$

- Parameter α is set to 1000.
- Slopes of the external PWL segments are set in a consistent way for all criteria, and assure that:
  - ◇ all CAFs are strictly monotone and concave, and
  - ◇ the slopes are neither too flat nor too steep, i.e., the corresponding optimization problem parameters do not cause numerical problems.

The CAF defined by (13) with the scaling transformation (14) conforms to all requirements; in particular, they provide appropriate (for numerical properties of the underlying optimization problems) scaling in criteria and achievement spaces. The scaling transformations are based on the aspiration and reservation values, which often differ orders of magnitude less than the utopia and nadir values. This property is justified by the practice and common sense, which show that for problems with large/huge range of values between utopia and nadir, the interesting analysis ranges (implied by the aspiration and reservation) is orders of magnitude smaller.

The CAF values have a very easy and intuitive interpretation in terms of the degree of satisfaction from the corresponding value of the criterion. Values of α and 0 indicate that the value of the criterion exactly meets the aspiration and reservation values, respectively. CAF values between 0 and α can be interpreted as the degree of satisfaction of the criterion value, i.e., to what extent this value is close to the aspiration level and far away from the reservation level. These interpretations correspond to the interpretation of the membership function from fuzzy set theory, which is discussed in [39]. In the latter, as well as in fuzzy sets, α is set to 1; for the reasons explained above we use much large values of α, but this does not change interpretation of CAF's values. In fact, the CAF extends the membership function concept because the CAF also takes negative values (for criteria values worse than the reservation), and values greater than α (for criteria values better than the aspiration). This extension is necessary for proper handling of any q^a and q^r values, which in turn frees the users from concerns regarding attainability of the considered aspiration and reservation levels.

3.3.2. Achievement Scalarizing Function

Achievement Scalarizing Function (ASF), denoted by $asf(\mathbf{caf})$, aggregates CAFs into a variable that serves as a GF of the optimization task providing Pareto solution fitting best the preferences specified by the user for the criteria. MCMA uses the established approach for $asf(\cdot)$ definition, see e.g., [20]:

$$asf(\mathbf{caf}) = \min_{i \in I_a}(caf_i(\cdot)) + \frac{\epsilon}{N} \cdot \sum_{i \in I} caf_i(\cdot), \tag{15}$$

where $caf_i(\cdot)$ is defined by (13), I_a and N denote the set of indices of active criteria, and the number of all criteria, respectively. The choice of criteria activity for each iteration is done by the user. All criteria are active by default; the user may, however, declare one or more criteria to be inactive. This is especially useful for problems with many criteria, when it is often desired to analyze trade-offs between subsets of criteria.

The main role of the ASF defined by (15) is to aggregate the CAFs; this is achieved by the first term. However, ASF defined by only the `min` term would not guarantee Pareto-efficient solutions. The latter is achieved by the second, regularizing term.

The max-min aggregation is motivated by the Rawlsian principle of justice [41] interpreted as preference for improving the situation (performance) of the weakest element (e.g., member of society or family). In the MCA context, it means improving the achievement of the worst performing criterion. This in turn implies that the ASF measures the overall achievement by the smallest value of **caf**; in practice, usually the worst CAF values are equal for two criteria.

The regularizing term guarantees an ϵ-properly Pareto-efficient solution. Formal explanation of this concept is beyond the scope of this paper; it can be found e.g., in [42]. Informally, it means that small (in terms indirectly defined by the ϵ-value) deviations of criteria value may be ignored when Pareto-efficiency is determined.

4. Modeling environment for MCMA

Multiple-criteria model analysis environment actually implemented and described in this paper is composed of three modular and interlinked parts:

- Modeling environment used for development and maintenance of the model instances.
- MCMA tool.
- Computational infrastructure.

Such architecture effectively exploits the proven modeling paradigms, combined with DBMS, XML, and the Web technologies, provides an efficient and robust implementation framework. The framework exploits encapsulation of the corresponding processes and enables use of MCMA tool with different modeling environments. Use of a modeling environment preferred for a specific problem has at least two obvious advantages:

1. The model development is encapsulated, i.e., model instances are developed in environment best suited to the needs of the developers. This also includes dedicated and/or adapted tools for the model verification and analysis.
2. Optimization solvers available in each environment are also used for solving optimization tasks of the MCMA.

The two-way linkage between model development environment and MCMA is composed of:

- Model instance conforming to the requirements summarized in Section 3.1 is interactively uploaded to MCMA.
- Solution of each MCMA iteration is provided to the user.

Currently two formats of model instances are implemented: the standard MPS-format, and two versions of the GAMS-formats (either a single file or a structured collection of GAMS specification and data files). Solutions are provided in formats corresponding to the model instance, i.e., either a standard MPS-format output file or the format defined in GAMS specification for the output.

The above outlined architecture has been proven by applications in diverse fields and modeling environments. The latter include model development tools dedicated for specific applications, e.g., modeling of agro-ecological zones [43], regional water quality management [44, 45], and regional air quality [46]. General purpose modeling environments were used for other MCMA applications, e.g., GAMS for assessment of nuclear power in global energy system [47], and GNU for the case study described in Section 5.

Discussion of model development environments is beyond the paper scope; it can be found in many publications, e.g., a comprehensive overview is provided in [30]. Further on we focus on the MCMA tool and follow with outlining configuration of the related computational infrastructure.

4.1. MCMA tool

4.1.1. Functionality from the user point of view

The implemented environment for MCMA also has modular structure. We focus its presentation on the functionality provided for users through the User Interface (UI), and only outline the underlying technology.

Following the SM paradigm also the MCMA process is structured and helps the user in effective and efficient analysis. The UI illustrated in Figure 5 provides users with flexible control of the all analysis elements, and thus enables organization of the process according to users' needs. The interface is available through Web-browsers, thus allows for the anytime-anywhere access; in particular, the analysis can be paused anytime; the defined tasks are anyway processed in background by servers, results stored and made available whenever the user decides to continue analysis.

We illustrate only one example of the interaction, namely the screen showing the distribution of solutions (in the left-side chart) and the control-panel for specification

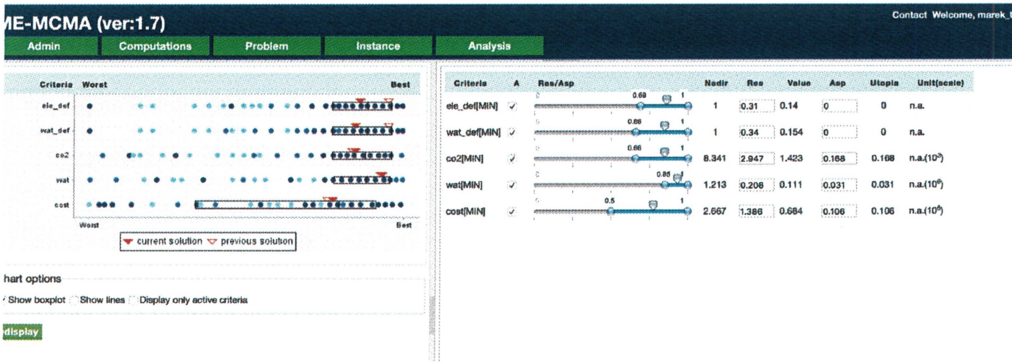

Figure 5: MCMA user-interface illustration.

of preference (in the right-side panel). Before discussing this main interaction, we summarize other MCMA functions controlled by the user through the green buttons (the corresponding screens are not shown):

Admin: Each MCMA user has private space for handling model instances and results of analysis. However, group of users may share their models and results. Therefore, MCMA supports administration of user groups, and privileges of group members.

Computations: MCMA provides information on the status of optimization tasks that have been generated but not yet finished, see Section 4.2 for explanation.

Problem: The corresponding set of dialogs supports uploading of model instances provided in one of the formats discussed above. The uploaded model instance is considered as the MCA problem.

Instance: For each the user may define several MCA instances. Each MCA instance is specified by the interactively defined criteria. The definition of each criterion is composed of selected:

- Corresponding outcome variables. The list of such variables is extracted from the uploaded model instance; filters for names of variables support selection for large models.
- Criterion type (either minimized or maximized).
- Criterion name. This is optional because the criterion name is initialized to be the same as the name of the corresponding outcome variable.

Analysis: The user may define for each MCMA instance several analyses. Each analysis is composed of iterations outlined above and discussed detail below. Defining several analyses is especially useful for extensive MCMA in which each analysis is composed of many iterations; moreover, separate analyses can have different focus and/or be done by different users. This issue is commented below.

Moreover, the white **Contact** button (at the top right corner of the blue control panel) provides link to the developers.

4.1.2. Preparatory computations

MCMA performs several background tasks in order to prepare for the user initial information for interactive analysis. The tasks are automatically generated and run after the user defines a new instance or a new analysis (the initial analysis is also generated automatically). We briefly summarize these background computations.

For each new instance the utopia and nadir values are computed. This requires $4 \star N$ automatically generated optimization tasks, where N is the criteria number. First N tasks compute the utopia values, for each criterion by the selfish optimization of the corresponding outcome variable. Next $3 \star N$ tasks sequentially improve approximation of the nadir values. After completing these tasks, the initial analysis is automatically created.

For each new analysis $N + 1$ iterations are automatically generated to provide the user with initial set of solutions. This set contains iterations in which only one criterion is active, and one iteration with the so-called compromise preferences, i.e., $\mathbf{q}^a$ and $\mathbf{q}^r$ set for each criterion at equal (in terms of the fractions of utopia and nadir range) values. Thus the user starts specification of his/her preferences with knowledge about the extreme (selfish-criterion) and compromise preferences. The example of the initial iterations is provided in Table 1 in Section 5.

4.1.3. Interactive specification of preferences

After the automatically generated iterations are completed, the user takes full control of further iterations. For each iteration the user analyzes the Pareto-solutions obtained in previous iterations, selects one solution as the basis for next iteration, and then considers which criteria he/she wants to improve and which should be compromised. Preferences for a new desired trade-off between criteria values are then expressed through aspiration and reservation values for each criterion, respectively. While defining new preferences one should consider that the basis solution is Pareto-efficient, i.e., an improvement of one criterion (or more criteria) is possible only, if at least one other criterion will worsen. An improvement of a criterion performance can be triggered by setting a more ambitious (closer to the corresponding utopia value) reservation value for this criterion, optionally augmented by also higher aspiration. Also optionally, one can select a criterion (or criteria) to compromise; this can be done by relaxing (i.e., worsening) the corresponding reservation value(s). In such a way the user preferences are defined for each iteration. For any given preferences, the multi-criteria problem is represented by an auxiliary parametric single-objective optimization problem defined through the achievement scalarizing function (15); solution of the corresponding optimization problem provides a Pareto-solution best fitting the user preferences.

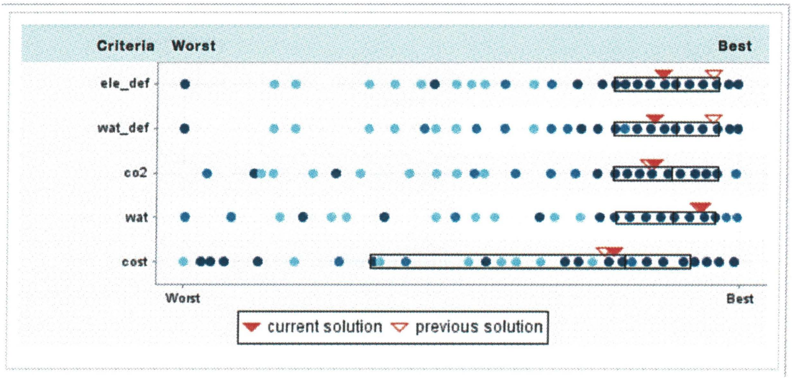

Figure 6: Distribution of criteria values.

The criteria values of the previously obtained (within the same analysis) solutions are presented in a chart composed of normalized parallel coordinates shown in Figure 6.

Typically, the MCMA users explore various areas of the Pareto frontier (e.g., cheap and expensive having the corresponding bad and good values of environmental criteria) before deciding which compromises between the criteria values fit best their preferences. Examples of such exploration are discussed in Section 5; more methodological background on the Pareto set analysis is available e.g., in [20, 39, 21].

4.2. MCMA computational infrastructure

The MCMA computational infrastructure is composed of:

UI: User interface application, implemented in Java, installed at a Tomcat servlet container, thus providing users with the MCMA interface through Web-browsers.

MCMA-solver: Dedicated solver, written in C++, transparent for the MCMA users; it manages most of the MCMA background tasks.

TM: Dedicated task manager; handles jobs generated and queued by MCMA-solver.

DB: Dedicated data-base, manages all persistent data of MCMA. We list below only examples of data to illustrate data scope:

- Configuration of the MCMA components; e.g., on solvers, functionality options available for diverse users and applications.
- Uploaded model instances, and their analysis.
- Status of all generated tasks.

- Users and user groups with privileges of members.

The DB also supports versioning. The schema of the MCMA DB is far too complex to be even outlined in this paper. We only point out that handling MCMA component configuration data through a DB greatly improves robustness and maintenance of this rather complex system.

Solvers: Set of optimization solvers distributed over the workstation network; the same solvers as used for the single criterion model optimization.

The workflows between elements of the MCMA infrastructure are actually hidden from the MCMA users, who interact primarily with the UI application. We briefly summarize the basic functions and dependences of the MCMA components.

The functionality related to uploading the model instance, specification of MCA problem instances, creation of analyses, and interactive analysis of solutions and specification of preferences is described above. Now we summarize flows and actions triggered by confirming preferences for each iteration:

1. Preference confirmation is done by the *Solve* request, which triggers storing the preference information and the iteration status in the DB; then the MCMA-solver is called, and the user may either wait for the solution, or switch to another iteration.
2. The MCMA-solver reads the iteration data from the DB, generates the MC-LP submodel, stores it on the server file-system, and updates the iteration status in the DB, which queues the corresponding optimization task.
3. The task manager is actually a daemon-type application, i.e., it runs it the background, frequently checks the queued tasks, and allocates their run on available servers and solvers.
4. Solvers are of three types:
   - Preprocessor: it merges the MC-LP with the model instance representation and generates input files for the selected optimizer, executes the suitable optimizer, waits until optimization finishes, and then calls the postprocessor.
   - Optimizers: solvers of the optimization problems.
   - Postprocessor: extracts from the optimization results solution of the MC-LP part, stores it in the DB, and calls back the MCMA-solver. The full solution remains available for the user.
5. MCMA-solver processes the solution, in particular prepares data for generation of the chart shown in Figure 5.

Each application updates the task status in the DB at the beginning and at the end of executions. Thus other application can check status and provide the user with the corresponding information, e.g., about execution stage for yet unfinished jobs, or charts and values for finished iterations.

4.3. Meeting the requirements

The requirements for the MCMA methods and implementation were implicitly defined by the following key attributes of the model and its analysis, summarized in Section 1. We summarize how each of them is addressed by the MCMA design and implementation:

1. Specification of the user preferences is done in easy interactive way. Preferences are specified in a natural way by aspiration and reservation values; such values have obvious meaning for the users familiar with the problem, and can be specified either as numbers (in the units used in the model specification) or by moving a slider, position of which corresponds to the degree (in term of utopia-nadir range) of reaching the utopia value. Moreover, the UI assures that A value dominates the R value. There are no other requirements for obtaining a Pareto solution. Therefore, the analysis can be effectively done also by users without modeling knowledge and skills.
2. Large ranges (several orders of magnitude) of criteria values are handled by appropriate internal criteria value scaling applied in the CAF specification, see Section 3.3.1.
3. Computations are organized in such a way models can be analyzed irrespective of the required computation time. The only condition is that the provided model instance can be solved for single-criteria optimization for each of the selected criterion. The same solver is used for MCMA and for the model development, and the computational complexity of single and multicriteria optimization is practically the same, see Section 4.
4. MCMA can supports users with diverse backgrounds and skills. The main part of MCA is typically done in the criteria space. This functionality is fully supported by the UI. The user only needs to iteratively specify preferences, click the *Solve* and wait for the results displayed in the same screen. For longer computations the optimization status is displayed. Users interested in details of the generation and runs of the corresponding tasks can use the provided links to full solutions and logs of all related activities.
5. MCMA uses a provided model instance; therefore, diverse modeling environments can be used for the model development. Thus, MCMA can be used in a concerted way with other diverse approaches to model analysis, including problem-specific post-optimization analysis and reporting.

5. Energy-climate-water nexus

5.1. The case-study problem and model

The interrelations of decisions on the energy and climate policies are commonly known and therefore do not require any comments here. Water availability is a key development factor in many regions; Saudi Arabia is a country facing water scarcity. Especially in such countries management of both energy and freshwater are required for meeting the development goals of societies.

Rational policy-making requires integrated energy and water systems planning, especially considering interrelations of these systems. Water plays a key role in the supply of energy in many regions globally, primarily for thermal power plant cooling and hydropower generation [48]. Constraints on the availability of water resources in these regions therefore pose risks to energy service reliability. At the same time, a significant amount of energy is required to extract, treat and distribute freshwater resources [49]. Constraints on the freshwater services supply therefore pose risks of additional energy requirements. These interdependencies promote integrated planning of water and energy infrastructure systems. Moreover, international commitments call for including in such planning also the climate impacts of the considered solutions.

The above outlined energy-water-climate nexus has motivated the model development described in [50]. Similar modeling activities appear to be needed for supporting policy-making also in many other countries.

5.1.1. The problem

The considered decision problem mainly deals with the infrastructure planning. Infrastructure here refers to the technologies or processes that enable energy and water services supply to consumers. Planning and design of regional energy and freshwater infrastructures involve a concerted choices of technologies and consideration of a wide variety of economic, social and environmental conditions, which makes it difficult to decide which technologies to invest in, and in what order. Exploration of rational combinations of technologies and level of investments requires appropriate methods and tools for development of the corresponding model and its analysis.

5.1.2. Outcomes and criteria

We present here only those model outcome variables that serve as criteria in the presented analysis case. In order to deal with a manageable number of criteria the corresponding variables represent the corresponding aggregations of the spatial and temporal problem dimensions, i.e., the planning horizon (2010-2050) and all sub-national regions (13 provinces).

The following five criteria were selected for the presented case:

Cost – the total investment cost (in 10^{12} US$) of the infrastructure transformation.

CO_2 – the total CO_2 emissions (in 10^9 metric tons).

Wat – the total groundwater extraction (in 10^3 km^3).

Ele_def – the maximum (over regions and time) domestic electricity deficit, i.e., shortage of covering the corresponding given demand (in fractions of the corresponding given demand, see (16)).

Wat_def – the maximum irrigation water withdrawals deficit i.e., shortage of covering the given demand).

The case with the first three criteria is presented in [50]. Here we present the analysis with the two additional criteria aimed at exploring how such a (controlled by the corresponding decisions) deficit can improve the performance in terms of the other three criteria. Specification of the first three criteria is straitforward therefore we don't discuss it. However, specification of the remaining two criteria desire short discussion.

Let $K = \{wat, ele\}$ denote the set of the deficit kinds, namely water and electricity, respectively, and R, T denote sets of regions and time-periods. The deficit criteria are defined by:

$$\text{Ele\_def} = \max_{r \in R,\, t \in T} \delta_{ele,r,t}, \qquad \text{Wat\_def} = \max_{r \in R,\, t \in T} \delta_{wat,r,t}, \tag{16}$$

where $\delta_{k,r,t} \in [0, 1]$ are decision variables defining fractions of the corresponding unmet demands.

5.1.3. The system-state variables

All five criteria defined above support analysis of outcomes aggregated over regions and time-periods. In order to augment such analysis by examination of consequences in specific regions and/or time-periods, the system-state variables were introduced. These variables represent such system properties as the chosen electricity-generation technologies, capacities of the power plants, water balances in specific locations, water transfers between regions.

As an example of state variables we define below the covered parts of the given electricity and water demands. The former three-criteria analysis assumed that all baseline projected demands (specified in [51]) were met. To enable analysis of controlled shortages (in terms of unmet demand of irrigation water and domestic electricity) the decision variables $\delta_{k,r,t}$ and two criteria defined by (16) were added. These criteria are also defined at the aggregated level. Therefore, other system-state variables represent the covered parts of the corresponding demands; these are denoted by $d_{k,r,t}$, and defined by:

$$d_{k,r,t} = b_{k,r,t} \cdot (1 - \delta_{k,r,t}), \qquad k \in K,\ r \in R,\ t \in T, \tag{17}$$

where $b_{k,r,t}$ stands for the baseline projected demands.

5.2. Key results of the analysis

We now present the key results of the MCA in the criteria space defined in Section 5.1.2. We start with discussing in Section 5.2.1 attributes of selected iterations. Then Section 5.2.2 overviews a sample of diverse solutions. We conclude the presentation of key results with Section 5.2.3 containing selected results in the system-state space.

5.2.1. Discussion of selected iterations

The common practice and the discussion in Section 3 show that a comprehensive analysis of any complex model involves many iterations. In this paper we can show only a small sample of iterations characterizing diverse criteria trade-offs. We will follow this by overview of the sub-set of a large number of iterations.

Table 1 presents criteria values for iterations selected to discuss a representation of the trade-offs between the criteria. The Table is composed of four parts discussed in a row below. The first (top) part provides the utopia and nadir values. The second part contains results of selfish optimization of each criterion, as well as so-called neutral solution. The remaining two parts present diverse iterations, all sorted by increasing cost.

	Cost		CO_2		Wat		Electr. deficit		Water deficit	
Utopia	0.11	(0%)	0.17	(0%)	0.03	(0%)	0	(0%)	0	(0%)
Nadir	2.67	(100%)	8.34	(100%)	1.21	(100%)	1.00	(100%)	1.00	(100%)
Utopia-3c	0.24	(9.4%)	0.46	(5.6%)	0.03	(0%)	0	(0%)	0	(0%)
Nadir-3c	2.67	(100%)	9.69	(118%)	1.26	(107%)	0	(0%)	0	(0%)
Iteration	**Cost**		**CO_2**		**Water**		**Electr. deficit**		**Water deficit**	
1599	0.11	(0%)	1.07	(11%)	0.28	(22%)	1.00	(100%)	1.00	(100%)
1991	0.55	(17%)	0.17	(0%)	0.13	(8%)	1.00	(100%)	1.00	(100%)
1992	1.82	(67%)	1.80	(20%)	0.03	(0%)	0.67	(67%)	0.67	(67%)
1989	0.88	(30%)	0.71	(17%)	0.13	(8%)	0.	(0%)	0.	(0%)
1680	0.52	(17%)	1.55	(17%)	0.23	(17%)	0.14	(17%)	0.14	(17%)
1669	0.14	(1%)	0.29	(2%)	0.28	(21%)	1.00	(100%)	1.00	(100%)
1982	0.23	(5%)	8.00	(98%)	1.21	(100%)	0.	(0%)	0.	(0%)
2008	0.24	(5%)	5.72	(68%)	0.84	(68%)	0.14	(14%)	0.14	(14%)
2007	0.25	(5%)	4.14	(49%)	0.61	(49%)	0.24	(24%)	0.24	(24%)
2010	0.25	(5%)	6.28	(75%)	0.92	(75%)	0.05	(5%)	0.05	(5%)
2011	0.29	(7%)	2.83	(31%)	1.09	(89%)	0.07	(7%)	0.07	(7%)
2023	0.29	(7%)	2.81	(31%)	0.88	(71%)	0.07	(7%)	0.32	(32%)
2022	0.30	(7%)	2.88	(32%)	0.43	(32%)	0.07	(7%)	0.65	(65%)
2004	0.30	(7%)	3.21	(37%)	0.58	(45%)	0.19	(19%)	0.19	(19%)
2005	0.31	(8%)	3.40	(39%)	0.50	(39%)	0.20	(20%)	0.20	(20%)
2014	0.31	(8%)	4.43	(52%)	0.65	(52%)	0.06	(6%)	0.06	(6%)
2006	0.32	(8%)	1.87	(21%)	0.28	(21%)	0.42	(42%)	0.42	(42%)
2012	0.32	(8%)	8.37	(39%)	0.48	(100%)	0.08	(8%)	0.08	(8%)
2015	0.36	(10%)	3.64	(42%)	0.54	(42%)	0.07	(7%)	0.07	(7%)
2013	0.42	(13%)	3.13	(36%)	0.42	(36%)	0.02	(2%)	0.02	(2%)
2009	0.43	(13%)	2.87	(33%)	0.43	(33%)	0.03	(3%)	0.03	(3%)
1999	0.57	(18%)	1.65	(18%)	0.25	(18%)	0.07	(7%)	0.07	(7%)
2003	0.58	(19%)	1.28	(15%)	0.19	(15%)	0	(0%)	0	(0%)
2000	0.62	(20%)	1.41	(16%)	0.21	(16%)	0.06	(6%)	0.06	(6%)
1718	0.62	(20%)	1.14	(16%)	0.21	(15%)	0.06	(6%)	0.06	(6%)
2002	0.68	(22%)	1.20	(13%)	0.18	(13%)	0.05	(5%)	0.05	(5%)
1736	0.68	(23%)	1.13	(12%)	0.17	(12%)	0.06	(6%)	0.06	(6%)
2001	0.72	(24%)	1.04	(21%)	0.16	(21%)	0.04	(4%)	0.04	(4%)
1595	0.77	(26%)	1.12	(12%)	0.14	(9%)	0.00	(0%)	0.00	(0%)
1696	0.77	(26%)	0.82	(8%)	0.13	(8%)	0.08	(8%)	0.08	(8%)
1700	0.78	(26%)	0.84	(9%)	0.13	(8%)	0.06	(6%)	0.06	(6%)
1676	0.78	(26%)	0.56	(5%)	0.20	(14%)	0.05	(5%)	0.05	(5%)
1701	0.79	(27%)	0.83	(8%)	0.13	(8%)	0.05	(5%)	0.05	(5%)
1627	0.83	(28%)	0.92	(10%)	0.13	(8%)	0.00	(0%)	0.00	(0%)
1786	0.92	(32%)	0.69	(7%)	0.11	(7%)	0.03	(3%)	0.03	(3%)
1610	1.27	(45%)	0.54	(5%)	0.08	(5%)	0.05	(5%)	0.05	(5%)
1784	1.66	(60%)	0.51	(4%)	0.08	(4%)	0.02	(2%)	0.02	(2%)
1682	2.56	(96%)	0.48	(4%)	0.08	(4%)	0.04	(4%)	0.04	(4%)
1691	2.67	(100%)	0.54	(5%)	0.04	(1%)	0.01	(1%)	0.01	(1%)

Table 1: Utopia and nadir criteria values, followed by values of criteria in selected iterations. Units for the criteria values: Cost [$\times 10^{12}$ USD], CO_2 [$\times 10^{9}$ metric tons], Wat [$\times 10^{3}$ km^3]. Values of the deficit criteria are defined as fractions, see eq. (16). The -3c suffix to utopia/nadir labels denotes the corresponding values of the three criteria analysis presented in detail in [50].

Criteria values are presented in the Table as pairs composed of the actual criterion value and the percentage of the utopia–nadir range (of the five criteria analysis instance). The percentage can be interpreted as a relative criterion optimality loss (compared to the corresponding selfish optimization), and therefore provides a good yard-stick for assessing individual criteria performance/goodness. Thus, 0% and 100% stand for the utopia and the nadir criterion values, respectively.

We start with commenting the first part of the Table. As discussed in detail in Section 3 the utopia and nadir values show the corresponding ranges of each criterion values for all Pareto-solutions. These ranges are huge (especially, if one considers also the corresponding measurement units shown in the Table caption), and thus call for comprehensive analysis of diverse Pareto-efficient solutions. Moreover, it is worth to note that the utopia values for criteria Cost and CO_2 for the five criteria instance are less than half of those for the three criteria problem instance, while the utopia value for the Wat criterion is the same for both instances. This shows the possibility of trade-offs between the Cost, CO_2, and the two deficit criteria, and no such trade-offs with the Wat criterion.

In the remaining discussion of Table 1 solutions are identified, for the brevity sake, by the #-character followed by the number (e.g., #1680 stands for iteration number 1680).

We move on with discussing the second part of solutions composed of selfish criteria optimization as well as the neutral solution. The first four iterations listed in this part show the selfish optimization results (these have the same criteria values for optimization of each deficit criteria, therefore only one of them is shown). Obviously, the optimized criterion reaches the corresponding utopia value at the expense of poor performance of at least one other criterion. It is worth to observe that performance of the other criteria is not always (very) bad (e.g., #1989). We recall that the latter is thanks to the regularizing term of the scalarizing function (15) that assures properly Pareto-solutions (see Section 3 for the explanations). Selfish optimizations very rarely provide acceptable trade-offs but often offer a good basis for exploring solutions focused on performance of the corresponding criterion.

The neutral #1680 solution is the last one generated automatically, and attempts to reach possibly balanced (in terms of equal relative performance of all criteria) solution. For our case-study it was possible to find a Pareto solution with equal (17%) performance of all criteria. From analytical point of view (e.g., in terms of the relative criteria performance defined above) such a solution might be considered as perfectly balanced. However, in actual decision-making it is not necessarily the best choice, e.g., because it might be rational to spend more money for achieving better values of at least some of the sustainability criteria. Therefore, such an analytical interpretation is typically not shared by all involved in the problem analysis. Nevertheless, #1680 provides another useful yard-stick. In particular, it shows that one cannot achieve performance better than 17% for all criteria simultaneously; it also indicates that at least one criterion shall reach a better value than 17%. Generally, the neutral solution often serves as a basis for starting various branches of analysis.

We complete the discussion of Table 1 by commenting on the results of the interactively generated iterations. Such iterations are the main part of the analysis. However, due to the space considerations we have to limit the discussion to a small number of selected key issues. The results of the selected interactive iterations are split into two parts, each composed of results with costs lower or higher than the cost of the neutral solution, respectively.

5.2.2. Overview of a sample of interactive solutions

Each Pareto-optimal solution has a certain trade-off (compromise) between criteria values that correspond to a given preference specification through a choice of the aspiration and reservation levels. Here we overview a rather large sample of iterations generated interactively within a specific analysis.

There are different and complementary ways of analyzing such a large number of diversified Pareto-efficient solutions. We start with commenting on the distributions of the corresponding criteria values shown in Figure 6 explained in Section 3. Note that the criteria values' distributions of the four criteria (Wat, CO2, Wat_def, and Ele_def) are very similar and substantially differ from such a distribution of the Cost criterion values. This corresponds to the bias of the prevailing preferences applied in the analysis, i.e., the focus on exploring solutions aimed at reaching the environmental goals represented by these four criteria in a cost-effective way. Therefore, the lowest distribution quartile of the costs (corresponding to the 25% of worst, i.e., expensive solutions) covers only about the third of the criteria value range while these quartiles of each of the other criteria values cover about 75% of the range. This shows that 75% of generated solutions have criteria values about either 25% (for all criteria but Cost) or 66% (for cost) worse than utopia. In other words, the preference bias was towards solutions rather reaching sustainability goals in cost-effective way than minimizing costs.

Further on in this section we comment on the iteration sub-set that excluded iterations similar in terms of the values of all criteria.

Results of the five criteria analysis are summarized in Figure 7 showing the normalized (in the same way as the values in Table 1) stacked criteria values. Worst normalized value is equal to 100. Therefore, the hight of each bar illustrates the cumulated performance of the corresponding iteration. It is not surprising that low-cost solutions have much worse cumulative performance of the other criteria, and expensive solutions have generally very good cumulative performance. However, it is interesting to note that solutions with practically same cost often have very different cumulated performance.

A complementary view on the cumulative criteria performance is shown in Figure 8, which shows the same set of solutions as Figure 7 sorted by the cumulative performance. Note that most solutions with low (up to 30) indices (i.e., at the left end of the chart) perform pretty well on all sustainability criteria and have moderate costs.

Figure 9 shows the same set of solutions as but sorted by increasing values of worst (within each solution) performing criterion. Here, the solutions are sorted by decreasing balance of the criteria performance. Such interpretation implies that a perfectly balanced solution has an equal performance of all criteria (note that this property has solution

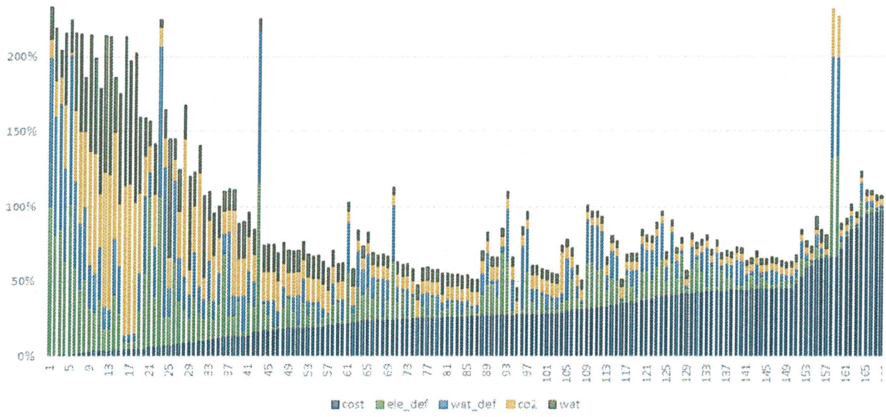

Figure 7: Values of all criteria sorted to increasing cost. Each bar stacks the normalized values of all criteria.

#1680 shown in Table 1).

5.2.3. Analysis in system state space

Impacts of the criteria settings on the provincial-level technology build-out for selected scenarios are provided in Figure 10. Depicted is the optimal annual electricity and freshwater supply mix in each region, as well as the interprovincial transfers and demand-levels. The cost-minimization solution (Figure 10a) involves expansion of relatively low-cost combined-cycle natural gas generation, with existing renewable energy policy driving development of 50 GW of mostly solar generation capacity. Groundwater withdrawals are left unconstrained in the cost-minimization model, and under the parameterized costs dominate the future water supply mix and displace existing interprovincial desalination transfers. Moreover, in the cost-minimization solution thermal power plants employ once-through freshwater cooling systems due to the low investment cost and lack of concern surrounding groundwater sustainability. The modeled extraction across sectors in this scenario likely exceeds available aquifer storage [52].

5.3. Lessons from the analysis

Water and energy systems are increasingly interdependent, and will benefit from integrated long-term development strategy. Multiple-criteria analysis support examination of trade-offs between attainable goals for diverse development objectives. In particular, the analysis shows diverse ways for reaching policy objectives in Saudi Arabia for 2050 that reduce cumulative groundwater extraction and electricity sector CO_2 emissions to levels likely needed to avoid local groundwater shortages and meet global climate stabilization targets are associated with a significant increase in system investment costs. The MCMA framework enables revealing a suite of solutions that remain nearly ambitious at

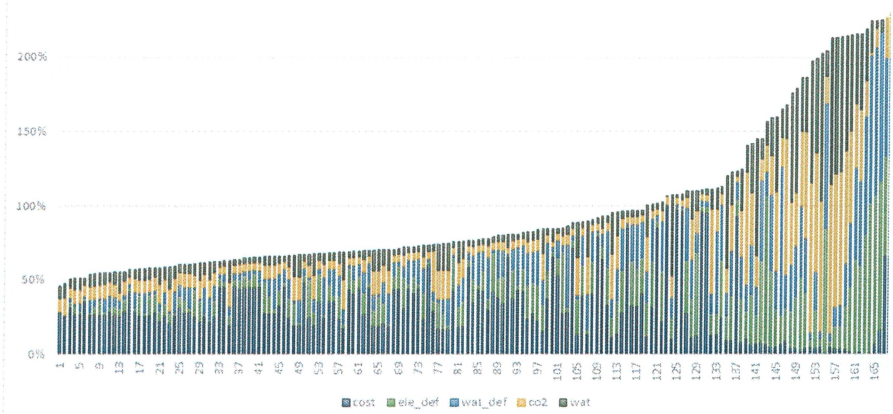

Figure 8: Solutions sorted by sums of criteria performance.

much lower costs. These savings would impact the the affordability of water and energy services in the rapidly developing nation of Saudi Arabia.

The presented approach and the corresponding model focus mainly on the electricity sector, and its interrelation with water and climate issues. Possible future work can extend this approach by two elements. First, to explore investments in technologies improving efficiency of electricity and water use. Then the currently defined electricity and water demand *deficits* can be modified to represent the corresponding decreases of the planned demands, which will result from the increased efficiency. Second, complementary to the first, to consider expanding the system boundaries to allow assessment from resource extraction through to end-use services. This would allow mapping the impacts from a more comprehensive set of technologies to energy and water sustainability metrics of interest. An important issue to address in this context is the linking of surface and groundwater management, which was simplified in the analysis due to surface water scarcity in the case study region. Moreover, the effects of other criteria important to regional planners (e.g., air pollution, energy security, investment risk, etc.) on the optimal development strategy should be explored to fully highlight potential trade-offs or synergies. The general MCMA framework described in this paper can readily be used for analysis of an extended model including these features.

6. Conclusion

Model-based support for decision-making in complex problems does, and will, require various elements of science, craftsmanship, and art (see, e.g. [21] for a collection of arguments that supports this statement). Science-based support for policy-making is a process, and quality of the support is determined by its weakest element. This paper focuses mainly on multicriteria analysis of attainable goals. The presented methods

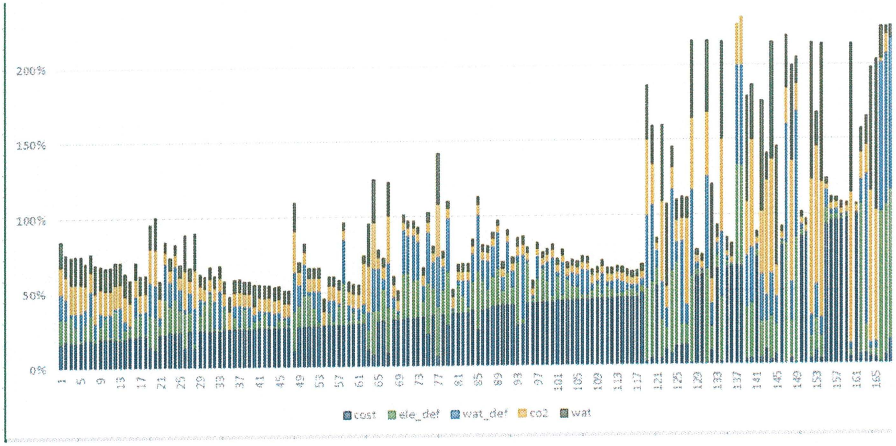

Figure 9: Solutions sorted by increasing values of the worst (within each solution) performing criterion.

and modular modeling environment contributes to meeting the needs of decision-makers and scientists. In particular, it enables multiple-criteria analysis of large-scale complex models developed in diverse environments. Moreover, such analysis can be effectively performed also by users without modeling background.

The presented actual decision problem on energy-water-climate nexus illustrates not only the effectiveness of the presented environment but also lessons from its analysis.

References

[1] H. Simon, A behavioral model of rational choice, Quarterly Journal of Economics 69 (1955) 99–118.

[2] H. Simon, Models of Man, J. Wiley & Sons, Chichester, New York, 1957.

[3] J. March, H. Simon, Organizations, J. Wiley & Sons, New York, 1958.

[4] H. Simon, Administrative Behavior, a Study of Decision Making Process in Administrative Organization, Macmillan, New York, 1958.

[5] H. Simon, Prediction and prescription in systems modeling, Operations Research 38 (1) (1990) 7–14.

[6] A. Wierzbicki, J. Wessels, The modern decision maker, in: Wierzbicki et al. [20], pp. 29–46, iSBN 0-7923-6327-2.

[7] D. Kahneman, A. Tversky, The psychology of preferences, Scientific American 246 (1982) 160–173.

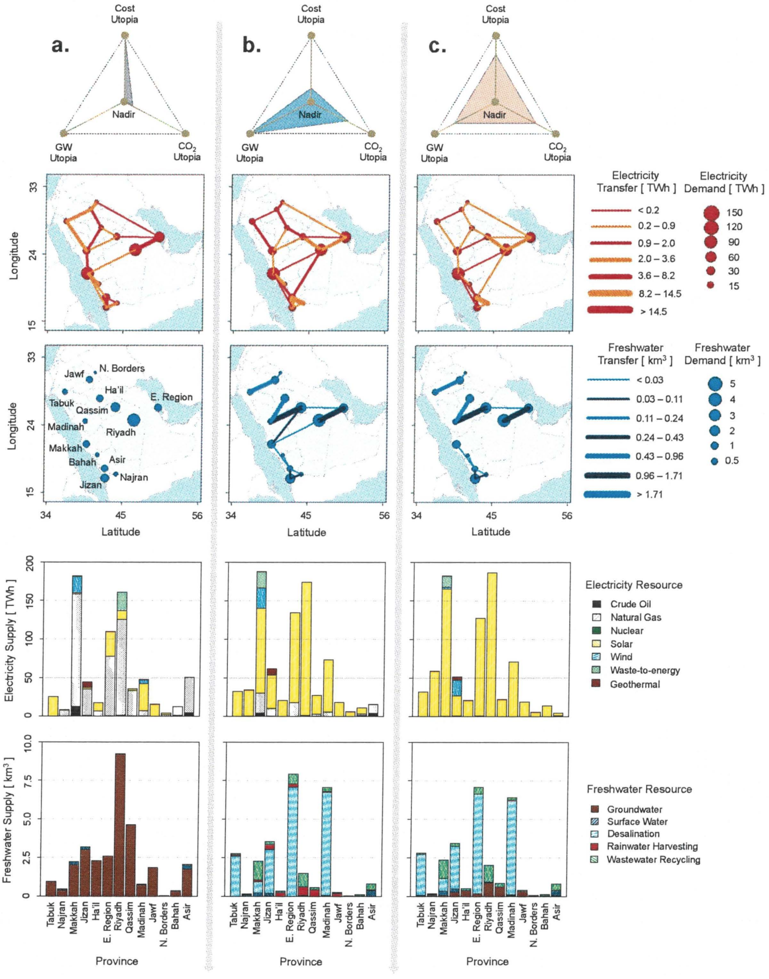

Figure 10: Provincial electricity and freshwater supply in 2050 for the selfish-optimization iterations listed in Table 1. a. Cost selfish (minimization) solution; b. Groundwater (GW) selfish solution; c. All criteria ambitious solution. The top row depicts the criteria outcomes in relation to the Utopia and Nadir points. Row two and three from the top depict the annual freshwater and electricity transfers between provinces, as well as the scale of annual demand. The bottom two rows depict the supply mix from the different resources.

[8] A. Tversky, D. Kahneman, The framing of decisions and philosophy of choice, in: G. Wright (Ed.), Behavioral Decision Making, Plenum, New York, 1985, pp. 25–42.

[9] H. Dreyfus, S. Dreyfus, Mind over Machine: The Role of Human Intuition and Expertise in the Era of Computers, Free Press, New York, 1986.

[10] A. Rapoport, Decision Theory and Decision Behaviour, Normative and Descriptive Approaches, Vol. 15 of Theory and Decision Library, Mathematical and Statical Methods, Kluwer Academic Publishers, Dordrecht, Boston, London, 1989.

[11] P. Yu, Forming Winning Strategies, An Integrated Theory of Habitual Domains, Springer Verlag, Berlin, New York, 1990.

[12] A. Rubinstein, Modeling Bounded Rationality, MIT Press, Cambrige, Massachusetts, 1998, iSBN 0-262-68100-5.

[13] R. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, J. Wiley & Sons, New York, 1976.

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

[15] J. Cohon, Multiobjective Programming and Planning, Academic Press, San Diego, 1978.

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

[17] R. Sprague, A framework for the development of decision support systems, in: Decision Support Systems: A Data Based, Model-Oriented, User-Developed Discipline, Petrocelli, Princeton, N.J., 1983.

[18] A. Lewandowski, A. Wierzbicki, Decision support systems using reference point optimization, in: A. Lewandowski, 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, 1989.

[19] Z. Pawlak, Rough Sets. Some Aspects of Reasoning about Knowledge, Kluwer Academic Publishers, Dordrecht, 1991.

[20] A. Wierzbicki, M. Makowski, J. Wessels (Eds.), Model-Based Decision Support Methodology with Environmental Applications, Series: Mathematical Modeling and Applications, Kluwer Academic Publishers, Dordrecht, 2000, ISBN 0-7923-6327-2.

[21] M. Makowski, A. Wierzbicki, Modeling knowledge: Model-based decision support and soft computations, in: X. Yu, J. Kacprzyk (Eds.), Applied Decision Support

with Soft Computing, Vol. 124 of Series: Studies in Fuzziness and Soft Computing, Springer-Verlag, Berlin, New York, 2003, pp. 3–60, iSBN 3-540-02491-3, draft version available from `http://www.iiasa.ac.at/~marek/pubs/prepub.html`.

[22] R. Ackoff, Management misinformation systems, Management Science 14 (4) (1967) 43–89.

[23] R. Ackoff, The future of operational research is past, Journal of Operational Research Society 30 (2) (1979) 93–104.

[24] W. Fisher, Utility models for multiple objective decisions: Do they acurately represent human preferences?, Decision Sciences 10 (3) (1979) 451–479.

[25] D. Maclean, Rationality and equivalent redescriptions, in: M. Grauer, M. Thompson, A. Wierzbicki (Eds.), Plural Rationality and Interactive Decision Processes, Vol. 248 of Lecture Notes in Economics and Mathematical Systems, Springer Verlag, Berlin, New York, 1985, pp. 83–94.

[26] C. Chapman, Science, engineering and economics: OR at the interface, Journal of Operational Research Society 39 (1) (1988) 1–6.

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

[28] A. Geoffrion, An introduction to structured modeling, Management Science 33 (5) (1987) 547–588.

[29] M. Makowski, A structured modeling technology, European J. Oper. Res. 166 (3) (2005) 615–648, draft version available from `http://www.iiasa.ac.at/~marek/pubs/prepub.html`.

[30] J. Paczyński, M. Makowski, A. Wierzbicki, Modeling tools, in: Wierzbicki et al. [20], pp. 125–165, iSBN 0-7923-6327-2.

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

[32] M. Ozturk, A. Tsoukias, P. Vincke, Preference modelling, Technical Report 2003-34, DIMACS, Rutgers University (2003).

[33] C. Chapman, My two cents worth on how OR should develop, Journal of Operational Research Society 43 (7) (1992) 647–664.

[34] M. Makowski, Management of attainable tradeoffs between conflicting goals, Journal of Computers 4 (10) (2009) 1033–1042, iSSN 1796-203X.

[35] A. Charnes, W. Cooper, Goal programming and multiple objective optimization, J. Oper. Res. Soc. 1 (1977) 39–54.

[36] M. Zeleny, A concept of compromise solutions and the method of the displaced ideal, Comput. Oper. Res. 1 (1974) 479–496.

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

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

[39] J. Granat, M. Makowski, Interactive Specification and Analysis of Aspiration-Based Preferences, European J. Oper. Res. 122 (2) (2000) 469–485, available also as IIASA's RR-00-09.

[40] A. Wierzbicki, The use of reference objectives in multiobjective optimization, in: G. Fandel, 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, 1980, pp. 468–486.

[41] J. Rawls, A Theory of Justice, Oxford University Press, London, Oxford, New York, 1973, iSBN 0-19-881301-5.

[42] A. Wierzbicki, Reference point methodology, in: Wierzbicki et al. [20], pp. 71–89, iSBN 0-7923-6327-2.

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

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

[45] M. Makowski, L. Somlyódy, River basin water quality management, Mathematical Modelling Theory and Applications 9 (2000) 311–332.

[46] M. Amann, C. Heyes, M. Makowski, W. Schöpp, An optimization model for the control of regional air quality modeling, in: J. Linders (Ed.), Modeling of Environmental Chemical Exposure and Risk, Kluwer Academic Publishers, Dordrecht, 2001, pp. 193–203.

[47] M. Lehtveer, M. Makowski, D. McCollum, M. Strubegger, Multi-criteria analysis of nuclear power in the global energy system: Assessing trade-offs between simultaneously attainable economic, environmental and social goals, Energy Strategy Reviews 8 (2015) 45–55, DOI 10.1016/j.esr.2015.09.004.

[48] IEA, Chapter 17: Water for Energy, in: World Energy Outlook, International Energy Agency, 2012, pp. 1–33.

[49] S. G. Rothausen, D. Conway, Greenhouse-gas emissions from energy use in the water sector, Nature Climate Change 1 (4) (2011) 210–219.

[50] S. Parkinson, M. Makowski, V. Krey, K. Sedraoui, A. Almasoud, N. Djilali, K. Riahi, A multi-criteria model analysis framework for assessing integrated water-energy system transformation pathways, Applied Energy 210 (15 January) (2018) 477–486, http://dx.doi.org/10.1016/j.apenergy.2016.12.142.

[51] S. C. Parkinson, N. Djilali, V. Krey, O. Fricko, N. Johnson, Z. Khan, K. Sedraoui, A. H. Almasoud, Impacts of groundwater constraints on Saudi Arabia's low-carbon electricity supply strategy, Environmental Science & Technology 50 (4) (2016) 1653–1662.

[52] S. Chowdhury, M. Al-Zahrani, Characterizing water resources and trends of sector wise water consumptions in Saudi Arabia, Journal of King Saud University-Engineering Sciences 27 (1) (2015) 68–82.