International Institute for
Applied Systems Analysis
Schlossplatz 1
A-2361 Laxenburg, Austria

Tel: +43 2236 807 342
Fax: +43 2236 71313
E-mail: publications@iiasa.ac.at
Web: www.iiasa.ac.at

Interim Report **IR-09-24**

Overview of Methods Implemented in MCA: Multiple Criteria Analysis of Discrete Alternatives with a Simple Preference Specification

Marek Makowski (marek@iiasa.ac.at)
Janusz Granat (J.Granat@itl.waw.pl)
Włodzimierz Ogryczak (W.Ogryczak@ia.pw.edu.pl)

Approved by

Detlof von Winterfeldt (detlof@iiasa.ac.at)
Director, IIASA

December 2009

Foreword

Practically all important decisions involve analysis of several (or even many), typically conflicting, criteria. Analysis of trade-offs between criteria is difficult because such trade-offs for most problems are practically impossible to be defined a-priori even by analysts experienced in Multi-Criteria Analysis (MCA). Therefore the trade-offs emerge during an interactive MCA which actually supports a learning process about the trade-offs. Hence, effective MCA methods are important for actual support of decision-making processes, especially those related to policy-making.

IIASA has been developing novel methods for MCA since mid 1970s, and successfully applying them to many practical problems in various areas of applications. However, there are new practical problems for which the existing MCA methods (developed not only at IIASA but also by many researchers all over the world) are not satisfactory. In particular, discrete decision problems with a large number of criteria and alternatives (the latter making pairwise comparisons by the users impracticable) demand new methods. For example, MCA analysis of future energy technologies involves over 60 criteria and over 20 discrete alternatives; a careful requirement analysis of this application has proven that none of the existing MCA methods is suitable for an effective analysis of the corresponding problem. Moreover, this analysis has been done by a large number of stakeholders with diverse backgrounds and preferences; most of them have no analytical skills, therefore the specification of preferences needed to be simple but still provide effective and intuitive analysis of the Pareto set.

The paper provides an overview of several new methods for MCA of discrete alternatives that have been implemented in the MCA, the Web-based application for multiple criteria analysis of discrete alternatives.

Abstract

Many methods have been developed for multiple criteria analysis and/or ranking of discrete alternatives. Most of them require complex specification of preferences. Therefore, they are not applicable for problems with numerous alternatives and/or criteria, where preference specification by the decision makers can hardly be done in a way acceptable for small problems, e.g., for pair-wise comparisons.

In this paper we describe several new methods implemented for a real-life application dealing with multi-criteria analysis of future energy technologies. This analysis involves large numbers of both alternatives and criteria. Moreover, the analysis was made by a large number of stakeholders without experience in analytical methods. Therefore a simple method for interactive preference specification was a condition for the analysis. The paper provides overview of several of new methods based on diverse concepts developed for multicriteria analysis, and summarizes a comparison of methods and experience of using them.

Acknowledgments

The authors gratefully acknowledge the long-term collaboration with, and help of Andrzej P. Wierzbicki of the National Institute of Telecommunications, Warsaw, Poland, who over three decades greatly influenced the developments of the multiple criteria methodology. The authors also thank Stefan Hirschberg, Peter Burgherr, and Warren W. Schenler, all of the Laboratory for Energy Systems Analysis, Paul Scherrer Institute, Villigen, Switzerland, for the numerous discussions during the over four years collaboration within the NEEDS Project.

The research reported in this paper was partly financially supported by the EC-funded Integrated Project NEEDS (project no: 502687), and by the Austrian Federal Ministry of Science and Research.

About the authors

Marek Makowski leads the IIASA Integrated Modeling Environment Project. His research interests focus on model-based support for solving complex problems, which incorporates three interlinked areas. First, integration of interdisciplinary knowledge and its representation by mathematical models. Second, creation of knowledge by comprehensive model analysis, including multicriteria methods. Third, tailoring the modeling process to meet the needs of decision-making processes. Thus Marek's research interests cover a cluster of areas relevant to the adaptation (whenever possible) or development (when needed) of methodology, algorithms, and software for model-based decision-making support. This includes more specific topics in Operations Research such as: multicriteria problem analysis, large scale optimization, optimization of badly conditioned problems, use of database management systems for complex models, decision analysis and support, user interfaces in decision support systems, effective treatment of uncertainty and risk.

Marek has published over 130 papers and book-chapters, co-edited four books, coordinated or led several scientific projects, and has been twice guest editor of the European Journal of Operational Research.

Janusz Granat is a leader of a Division of Advanced Information Technology at the National Institute of Telecommunications. He also lectures on decision support systems and management information systems at the Warsaw University of Technology. His scientific interests include decision support systems, multi-criteria analysis, modeling, data mining, event mining, techno-economic analysis and the design of the telecommunications network. He has been involved in various industrial and scientific projects e.g., data warehousing and decision support systems for telecommunication industry, building data mining models for marketing departments, development of decision support systems for energy management.

Włodzimierz Ogryczak is a Professor and Deputy Director for Research in the Institute of Control and Computation Engineering at the Warsaw University of Technology, Poland. His research interests are focused on models, computer solutions and interdisciplinary applications in the area of optimization and decision making. Within these the main stress is on the methodology of multiple criteria optimization and decision support, decision making under risk, location and distribution problems. He has published three books and numerous research articles in international journals.

Contents

List of Tables

List of Figures

Overview of Methods Implemented in MCA: Multiple Criteria Analysis of Discrete Alternatives with a Simple Preference Specification

Marek Makowski[*] *(marek@iiasa.ac.at)*
Janusz Granat[**] [***] *(J.Granat@itl.waw.pl)*
Włodzimierz Ogryczak[***] *(W.Ogryczak@ia.pw.edu.pl)*

1 Introduction

The paper has a rather technical character; it provides an overview of several new methods for Multiple Criteria Analysis (MCA) of discrete alternatives that have been implemented in the MCA, the Web-based application for multiple criteria analysis of discrete alternatives; the user guide and tutorial to the MCA is available in [6].

This report is addressed to advanced users who are interested in general properties of the new methods. Detailed presentation of some of these methods and their methodological background together with several associated concepts, including automated pairwise comparisons which lead to the corresponding pairwise outperformance aggregations, are provided in [2].

The structure of the paper is as follows: Section 2 summarizes the background of the MCA, followed by the simple approach to specification of preferences and their representation is solvers described in Section 4 and 5, respectively. The next two Sections provide specifications of objects and functions used in the specifications of methods. The methods are specified in Section 8. Section 9 summarizes results of comparing the methods. The remaining two Sections provide auxiliary information, and the Appendix summarizes the basic information about the Lorenz curve and quantile measures.

2 Background

Multi-Criteria Analysis (MCA) deals with finding optimal (best in the sense of selected goal function) solution[1] for a problem characterized by a vector of outcomes;[2] i.e., the

[*]Integrated Modeling Environment Project, IIASA.
[**]National Institute of Telecommunications, Warsaw, Poland.
[***]Warsaw University of Technology, Warsaw, Poland.

[1]For discrete problems alternatives are usually called alternatives; in this paper we use both terms as synonyms.

[2]Outcomes are often called attributes, or criteria, or indicators. We use these terms interchangeably, and also often omit the phrase *vector of*, i.e., outcomes actually stands for vector of outcomes.

problem can be formulated as a multiple-criteria optimization one. Without a loss of generalization one usually considers maximization of all outcomes. The qualitative difference between the classical (single-criterion) optimization and the MCA is that for the former one it is possible to define an apriori goal function which induces a complete ordering in the solution space, thus allows for finding an optimal solution.

For a truly multi-criteria problem it is impractical to define such an apriori goal function, therefore there exist alternatives that cannot be ordered (based on a formal comparison resulting in deciding that solution is better than another one). In many applications it is possible to limit the analysis to a subset of all solutions called the *Pareto set*.[3] A solution is called Pareto-efficient, 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 cannot improve any criterion without worsening at least one other criterion. Solutions that are not Pareto efficient are called dominated.

Alternatives belonging to a Pareto set cannot be compared in mathematical sense, i.e., one cannot objectively decide which one of any two selected from this set is better. However, the user has to eventually select one alternatives at the *best* one. The user is usually able to make a pair-wise comparison, i.e., subjectively select one of two presented alternatives as a preferred one. Pair-wise comparisons can be very efficient, if the numbers of both alternatives and criteria are very small.[4] For non-trivial problems however computer-based support is needed for helping the user to specify preferences in a structured way, and to modify them while learning about attainable trade-offs between the criteria values. The preferences are typically specified in two categories:

- a measure of satisfaction level from achieving specific values of each criterion;
- information on trade-offs between the satisfaction levels of different criteria.

This document summarizes methodological background and outlines implementation of several MCA methods developed for a specific class of multicriteria analysis of a set of discrete alternatives characterized by:

- large number of discrete alternatives (about 20), each defined by about 40 attributes (some of the attributes having multimodal distribution of values);
- large number of criteria (about 60) organized in hierarchical structure (the criteria include the attributes that define the alternatives);
- large number of stakeholders (about 3000) invited for making individual analysis; the stakeholders have diversified backgrounds, very few of them have knowledge/experience about/in multicriteria analysis.

We stress that we deal with multicriteria analysis (as opposed to the commonly used term multicriteria decision analysis). This class of problems requires:

- development of new MCA methods (see [1] for the justification of this statement), and also clarification of several methodological issues;
- a simple and intuitive interface for specification of user preferences that meets the requirements specified in [7].

These both elements may be interesting for a broad audience of researchers and practitioners involved in the MCA.

[3] Also called: Pareto-efficient solutions, Pareto frontier, non-dominated solutions. 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 [13].

[4] Note that 15 pair-wise comparisons are needed for 6 alternatives.

In order to help the users to analyze the problem and to find the *best* alternative we have implemented a Web-based interactive iterative procedure composed of the following steps:

- the user specifies preferences in the criteria space;
- the preferences are used for defining an ad-hoc goal function, and applying it for finding a solution;
- the user confronts the criteria values in the found solution with the specified preferences;
- the moves to the first step as long as the trade-offs between the found criteria values differ from the user expectations.

Thus we consider MCA as an iterative process composed of iterations. The first iteration is generated automatically. Then the user can select any of the iterations (from the tree of iterations belonging to his/her analysis composed of the initial iteration, and the iterations he/she has made) as a basis for creating a new iteration. Upon analysis of criteria values for the selected basis iteration the user modifies her/his preferences, and calls the solver, which provides a Pareto-efficient alternative corresponding best to the specified preferences.

This typical MCA procedure contains the main challenge for any multicriteria analysis method: *how to find a Pareto solution that matches best the user preferences*. This challenge appears to be even bigger, if the MCA methods are designed and implemented for users having no experience in mathematical modeling. However, it has been observed that also users with extensive experience in MCA have problems with a consistent specification of preferences, if such a specification involves a process with several interlinked steps.

Specification of preferences for all methods described in this paper is done in the probably easiest way: by selecting a relative importance of each criterion. The relative importance is expressed in qualitative terms, and then the relative importance are mapped (see Section 6.3) into the user defined weights. Note that the weights are normalized, therefore specifying an equal importance for all criteria has the same effect irrespectively of the selected importance level.

Thus the user forms a pattern of relative criteria importance, and expects to get a Pareto solution (alternative) having the corresponding pattern of criteria values, e.g., possibly best values of all very important criteria, good values for all important criteria, etc. However, usually there is no alternative having criteria values that correspond well to the user expectations.

The MCA is actually a learning process during which the user modifies his/her preferences in order to find an alternative with trade-offs between criteria values that fits best the user preferences. In other words, a specification of preferences is a tool for finding a preferred alternative. The preferred means that it has the best (in subjective opinion of the user) trade-offs of the criteria values amongst all alternatives; this should not be confused with the specified trade-offs which are often not attainable (i.e., there exists no alternative having the specified pattern of criteria values). Moreover, it is not really important which pattern of criteria importance led to finding the preferred alternative; actually, each alternative can be found for many rather different specified preferences.

A good method for MCA shall therefore provide the user with an intuitive way to modify specification of preferences in order to help the user to find his/her preferred alternative. The user preferences are then used as parameters for a selected Scalarizing

Function SF that induces an ad-hoc complete order in the set of Pareto alternatives. In other words, the SF is defined on vectors of outcomes (characterizing each alternative), and thus assigns a real value for each alternative. Thus any MCA method can be characterized by the parameters representing the user preferences, and the corresponding SF (or a procedure used instead of the SF for inducing a complete order in the alternative space).

A SF is actually a way for aggregating the individual outcomes according to the user preferences. SFs have various forms and properties that correspond on diversified approaches to preference modeling. Nevertheless, most scalarizing functions can be viewed as two-stage transformations of the original outcomes.

First the individual outcomes are rescaled to some uniform measures of achievements; these measures may include some preference parameters. Thus, the role of individual achievement functions is to measure (on a scale common to all criteria) the user satisfaction level related to each possible criterion value. This measure does not concern preferences related to trade-offs between criteria.

In the second stage the individual outcomes transformed into a uniform scale of individual achievements they are aggregated through a definition of the SF into a final scalarization. Different MCA methods use different aggregations (e.g., measuring the total or average, or the worst individual achievement). Moreover, the methods differ by one of two possible assumptions. The first assumes that the aggregation is impartial or symmetric with respect to the individual achievements; thus the achievements are treated as equally important. The second approach uses for the aggregation diversified measures reflecting relative criteria importance.

3 Structure of the multicriteria analysis specification

MCAA is an iterative process, i.e., the user makes a sequence of iterations for a selected analysis instance. For efficiency reasons analysis instances are defined in three stages:
1. problem specification (Sec. 3.1)
2. problem instance specification (Sec. 3.2)
3. analysis instance specification (Sec. 3.3)

For each analysis instance (later on called *analysis*) initial iteration is generated automatically. Then the user selects any iteration (from the tree of iterations composed of the initial iteration, and the iterations he/she has made), analyses the current trade-offs between criteria, specifies new preferences, and calls a solver. Thus an iteration is composed of:
- specified preferences for the criteria (equitable preferences are assumed for the initial iteration),
- the corresponding Pareto solution,
- optionally,[5] a method chosen for selecting a Pareto solution.

3.1 Problem specification

Multicriteria problem specification is composed of three parts:
- attribute names[6], indexed by $i = 1, \ldots, n$ (also denoted $i \in I$).

[5] A default method is used, if the user does not select a method.
[6] Attributes are also called indicators or outcomes, or criteria.

- alternative names, indexed by $j = 1, \ldots, m$ (also $j \in J$),
- attribute values denoted by q_{ij} specified for each pair $\{i, j\}$

Thus the attribute values are organized in a matrix $\boldsymbol{Q}$ composed of elements q_{ij}. For convenience j-th column of $\boldsymbol{Q}$ (composed of values of all criteria for j-th alternative) is denoted by $\boldsymbol{q}_j$, and i-th row of $\boldsymbol{Q}$ (composed of values of i-th criterion for all alternatives) is denoted by $\boldsymbol{q}_i$.

3.2 Instance specification

Multicriteria problem instance is composed of:
- Set of alternatives (selected from the problem specification).
- Set of criteria defined using a selected attribute and one of defined types (maximized, minimized, target[7]).
- Optionally, a hierarchy of criteria (Sec. 3.2.1) can be defined. In such cases, when it is necessary to distinguish the criteria derived from attributes they are called leaf-criteria or lowest-level criteria.

3.2.1 Hierarchy of criteria

Optionally, criteria can be organized in a hierarchical structure forming a tree, as illustrated[8] in Fig. 1.

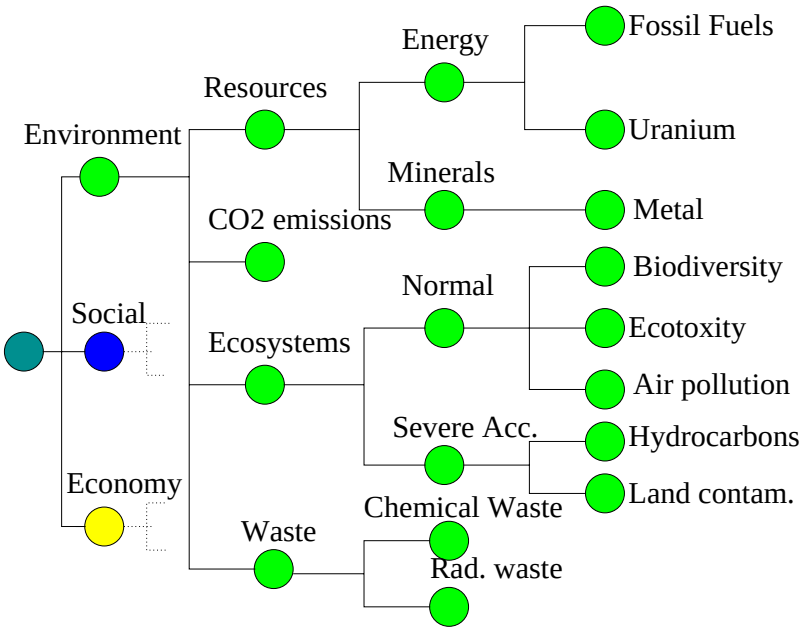

Figure 1: Example of criteria hierarchy.

Criteria values (Sec. 3.2) are specified only for leaf-nodes (lowest-level criteria). However, preferences are specified for all leaf-criteria, and for those intermediate nodes (i.e.,

[7] Interface to the target-type has not been implemented yet.

[8] Only one branch from the top level criteria in shown. Therefore criteria belonging to the Economy and Social criteria sets are not displayed.

excluding the root node) for which they can influence the selection of Pareto solutions.

We stress that the analysis results can be interpreted only for the leaf-criteria. Higher level criteria are used only as an additional way of specification of preferences for the leaf-criteria (see Section 6.3). Actually, one can always receive the same solution for two associated problem instances:

- With a criteria hierarchy by specifying the preferences for all criteria; the preferences for higher (above the leaf) levels are then used for modifying the preferences at the leaf level, see Sec. 6.3 for explanations.
- Without the hierarchy; the preferences are specified only for the leaf-criteria (the only criteria for which attribute values are known).

3.3 Analysis specification

An analysis of the problem instance is – from the user of point of view – just a container for a set of iterations done for by a user for a selected problem instance. This approach provides a structure for possibly large sets of iterations, and also to assure privacy (the user of the MCAA has access only to iterations they he/she has created).

First iteration is generated automatically (with equal preferences for all criteria selected for the corresponding model instance). Each user created iteration starts from a previously done iteration that is selected by the user. Therefore the user implicitly creates a tree of iterations. To create a new iteration the user:

- Selects a parent iteration.
- Selects a MC method (from the set of methods described in Sec. 8), if more then one method is provided for the user.
- Specifies the preferences (Sec. 4).
- After specifying the preferences clicks on the *Solve* button.

  For each selected iteration the user may:[9]
- Attach a note.
- Examine the chart of alternative characteristics.*
- Examine the chart comparing alternatives by criteria.*
- Browse through the problem instance data.[10]*
- Users having access to more than one analysis can browse (by clicking on the *Return* button) the tree composed of: problems, instances, and analyses.
- Browse through the help.*

4 Specification of preferences

4.1 Required for all methods

- ri - relative importance of criteria, specified for each criterion by a button position; currently 8 buttons (numbered by 0 through 7):

[9] Each of the actions marked below by the * characters results in opening a new window. Each of these windows (except of the help) is labeled by the a string composed of: problem name, instance name, analysis name, iteration number, and the symbol of the MC method used for this iteration.

[10] The data are of course common for all iterations, however the access is provided from each iteration.

⋆ 0-th button: ignore the criterion;
Note: criteria "below" (i.e., children, grandchildren, ...) ignored criteria are assumed to be also ignored (therefore solvers redefine the specified values of corresponding ri_i to 0).
⋆ 4-th button: average importance;
⋆ buttons 5 through 7: more, much more, vastly more, important than average, respectively;
⋆ buttons 3 through 1: less, much less, vastly less, important than average, respectively;

4.1.1 Required for some methods

- $\boldsymbol{impr}$ - selection of criteria that shall be improved and those to be compromised; this is specified for each criterion by a button position; currently 4 buttons (numbered by 0 through 3):
  ⋆ 0th button: allow to compromise (worsen) the criterion value;
  ⋆ 1st button: free the criterion (change in any direction);
  ⋆ 2nd button: stabilize the criterion value (preference for keeping changes small);
  ⋆ 3rd button: improve the criterion value;

4.1.2 Optional or computed from data

- $\boldsymbol{res}$ reservation and $\boldsymbol{asp}$ aspiration values for each criterion
- $\boldsymbol{rfp}$ reference point (one value for each criterion)

5 Representation of preferences in MC solvers

The preferences are specified for each iteration (except of the initial one). First, the user optionally selects for each iteration the method which will be used to find a Pareto solution that fits best his/her preferences. The way the preferences are specified depends on the method so advanced users may experiment with different methods and find the favorite one. Each method uses the associated solver. The methods/solvers differ by the internal representation of user preferences, and the way in which a Pareto solution is selected for specific preferences. However, several elements of solvers are common, and are therefore presented before each method will be outlined with method-specific elements.

All methods use the following six[11] types of objects and corresponding functions:

- selection of active leaf-criteria
- selection of Pareto alternatives
- $w_i(\boldsymbol{ri})$ and $v_i(w_i)$ - criteria scaling/weighting;
- $IA_i(\boldsymbol{q_i})$ - Individual Achievement functions measuring (for each criterion separately) the satisfaction level corresponding to a value of the criterion;
- $AF_i(\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{IA})$ - Achievement Function measuring (for each criterion) the satisfaction level corresponding to a value of the criterion taking into account relative importance (represented by $w_i(\boldsymbol{ri})$ or/and $v_i(w_i)$) of all criteria;
- $SF(\boldsymbol{AF})$ - Scalarizing Function measuring satisfaction levels for each alternative.

[11] We present here only a subset of solver elements that are necessary for understanding the methods used.

The first four are common for all methods, while the other two are specific for a method (or a set of methods). For the latter some auxiliary functions or relations are defined later. Therefore we first (Sec. 6) define the common functions, and then (Sec. 8) introduce methods, each of the latter accompanying with the corresponding definitions of $\boldsymbol{AF}(\cdot)$ and $\boldsymbol{SF}(\cdot)$.

6 Objects and functions common for all methods

6.1 Active leaf-criteria

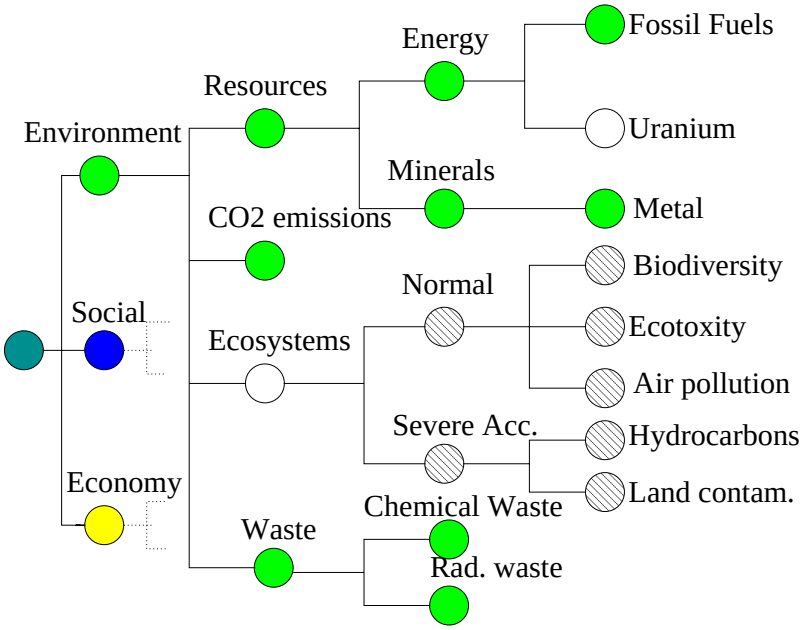

Figure 2: Hierarchy of active criteria

The user may chose to ignore some criteria (cf Sec. 4.1) therefore the set of active leaf-criteria has to be defined for each iteration. This is a trivial operation for analysis without hierarchical structure. If criteria hierarchy is defined (see example in Fig. 1) then criteria at any hierarchy level can be specified as inactive, see example in Fig. 2. In such cases then the set of active-leaf criteria is defined as follows:

- full criteria tree is defined (see Fig. 1).
- Activity of all criteria is defined according the selection of the value of relative criteria importance button (Sec. 4.1).
- active criteria tree is defined by removing from the full criteria tree nodes corresponding to inactive criteria and the branches originating from such nodes.[12] Note that activity of criteria imply changes in the impact of the selected relative criteria importance, see Sec. 6.3.

[12]In the example shown in Fig. 2 only two criteria (marked by white nodes) were selected to be not active. However seven more criteria (marked by gray nodes) become inactive because their parent criteria are inactive.

• The set of active leaf-criteria is composed of leaves of the active criteria tree.

Note: Further on we use the term *criteria* for the active leaf-criteria, since only such criteria are considered for analysis. Also the number of criteria n denotes the number of active leaf-criteria. The role of intermediate-level active criteria is defined in Sec. 6.3.

6.2 Set of Pareto alternatives

The set of Pareto alternatives[13] is defined for the considered criteria according to the commonly used definition: *A solution is called Pareto-efficient, 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 cannot improve any criterion without worsening at least one other criterion. Solutions that are not Pareto efficient are called dominated.

Note that each analysis iteration is actually composed of a series of subproblems defined for determining ranking of alternatives (see Sec. 10). For each subproblem a new set of Pareto alternatives is defined.

6.3 Criteria relative importance

Preferences for all but one (described in Sec. 8.1.1) methods described in this note are specified as relative criteria importance $\boldsymbol{ri}$, see Sec. 4.1. The $\boldsymbol{ri}$ are mapped into two associated vectors:

• $w_i, \quad i = 1, \ldots, n$

• $v_i, \quad i = 1, \ldots, n$

Definition of criteria scaling coefficients (traditionally called weights) $\boldsymbol{w}$ is rather complex and therefore before presenting it we specify the simple definition of $\boldsymbol{v}$ composed of two stages: First, components of $\boldsymbol{v}$ are defined as:

$$v_i = 1/w_i, \qquad i = 1, \ldots, n. \tag{1}$$

Second, the $\boldsymbol{v}$ is normalized using the standard procedure:

$$v_i = v_i / \sum_{i=1}^{n} v_i, \qquad i = 1, \ldots, n \tag{2}$$

The definition of $\boldsymbol{w}$ is done in two stages:

1. Relative criteria importance (for all active criteria) are mapped into real values using one of the approaches specified in Sec. 6.3.1 and Sec. 6.3.2, respectively. The choice of the mapping is specific for the method.
2. Procedure described in Sec. 6.3.3 is applied, if criteria hierarchy is defined.

6.3.1 Standard mapping

For the current implementation (seven importance levels for not ignored criteria) the standard mapping is defined by:

$$w_i = ri_i/6, \qquad i = 1, \ldots, ncrit \tag{3}$$

[13] Also called: Pareto-efficient solutions, Pareto frontier, non-dominated solutions. For the sake of brevity we don't deal here with more advanced concepts, e.g., properly efficient solutions.

where $ncrit$ is the number of all active criteria, and ri_i is the position of the selected button position indicating relative importance of i-th criterion.[14] After the mapping the vector $\boldsymbol{w}$ is normalized in the same way as (2).

6.3.2 Multiplicative mapping

Multiplicative mapping is less popular than the standard mapping, but it has a number of advantages (see, e.g., [3]) therefore it is used by almost all methods. The mapping is defined by:

$$w_i = (\sqrt{2})^{x_i} \tag{4}$$

where x_i is selected from the ri_i-th position from the following vector:

$$\{-8, -4, -2, 0, 2, 4, 8\} \tag{5}$$

In other words, the values of weights can be selected from the vector

$$\{1/16, 1/4, 1/2, 1, 2, 4, 16\} \tag{6}$$

from the position defined by the relative importance button.

After the mapping the vector $\boldsymbol{w}$ is normalized in the same way as (2).

6.3.3 Weights for criteria hierarchy

If a criteria hierarchy is defined then (after one of the above described mappings is applied to all active criteria) the following procedure is applied:

1. Denote by $\boldsymbol{wa}$ weights defined for all active criteria
2. Define sets $S_k, k = 1, \ldots, K$ composed of siblings (i.e., nodes having a common parent node) of active criteria.
3. Normalize subsets of siblings:

$$wa_l = wa_l / \sum_{l=1}^{L_k} wa_l, \qquad l \in S_k, k = 1, \ldots, K \tag{7}$$

   where L_k is the number of elements in S_k.
4. For each leaf-criterion define

$$w_i = \prod_{k \in M_i} wa_k, \qquad i = 1, \ldots, n \tag{8}$$

   where set M_i is composed of indices of the following active criteria:
   - i-th leaf criterion
   - intermediate-levels criteria belonging to the branch of the active criteria tree leading to the i-th criterion.
5. Normalize $\boldsymbol{w}$:

$$w_i = w_i / \sum_{i=1}^{n} w_i, \qquad i = 1, \ldots, n \tag{9}$$

[14] Note that ignored criteria are not included, therefore $w_i > 0$.

6.4 Individual Achievement Functions

For each criterion define the default $IA_i(\boldsymbol{q_i})$ (Individual Achievement) function is defined for each criterion separately. $IA_i(\boldsymbol{q_i})$ measures the level of satisfaction (goodness) for each possible value of the corresponding criterion.

IA functions have the same properties as the Achievement Function in the RFP (Reference Point) method[15], in particular to be strictly monotone (increasing/decreasing for maximized/minimized criteria.[16]). However, in the RFP methods the Achievement Function represents also the inter-criteria preferences. Therefore in the current implementation we distinguish the ***IA*** and the ***AF*** functions, both having the same mathematical properties, but the latter including inter-criteria preferences.

IA are defined as follows:

1. Denote (for each criterion) by utopia and nadir the best and the worst (over all alternatives) values of the criterion.
2. Set the default values of asp_i (aspiration) and res_i (reservation) to be equal to the corresponding utopia and nadir values.
3. Implicitly define the default $IA_i(\boldsymbol{q_i})$ function[17] to be a PWL (Piece-Wise Linear) function composed of one segment:
   - $IA_i(nadir_i) = IA_i(res_i) = 0$
   - $IA_i(utopia_i) = IA_i(asp_i) = 1$
   - $IA_i(\cdot) \in [0, 1]$:

   Note: Such an $IA(\cdot)$ are equivalent to the normalized and unified (to make all **maximized**) criteria value mappings y_{ji} in the standard implementations of the weighted-sum methods, which typically define y_{ji} as:
   - scaled and shifted to the range $y_{ji} \in [0, 1], j \in J, i \in I$
   - maximized; i.e. a minimized $\bar{y}$ is replaced by $y = 1 - \bar{y}$.

   Therefore the default values of utopia, aspiration, reservation, nadir in the weighted-sum method are defined implicitly as:
   - $utopia_i = asp_i = 1$
   - $nadir_i = res_i = 0$

7 Auxiliary functions

The following functions are used by more than one method, therefore we defined them here.

7.1 Ordered Achievement Functions (OAF)

Ordered achievement function values $\boldsymbol{OAF}_j(\cdot)$ are defined for each alternative as sorted (in order corresponding to improving the argument values[18]) of $AF_i(\boldsymbol{q}_j)$, where $\boldsymbol{q_j}$ is the

[15] Actually also implicitly used by the WS (Weighted-Sum) method.

[16] Target-type criteria are handled by $IA(\cdot)$ composed of two $IA(\cdot)$: maximized/minimized for criteria values smaller/larger than the specified target value.

[17] By $\boldsymbol{q_i}$ we denote vector composed of i-th criterion values (for all alternatives).

[18] Equal values are ordered randomly.

vector of criteria values for j-th alternative):

$$\boldsymbol{OAF}_j = \mathrm{sort}(AF_i(\boldsymbol{q_j})), \qquad j \in J \tag{10}$$

where $AF_i(\cdot)$ is the achievement function specific for each method.

7.2 PWL functions for Quantile Aggregation

Define a PWL (Piece-Wise-Linear) function generated by vector of $n+1$ points $\{\boldsymbol{x}, \boldsymbol{y}\}$:

$$x_i = i/n, \qquad y_i, \qquad i = 0, 1, \ldots, n \tag{11}$$

where $\boldsymbol{y}$

$$y_0 = 0; \qquad y_i = y_{i-1} + \alpha s_i \qquad i = 1, \ldots, n \tag{12}$$

where $\boldsymbol{s}$ denotes preferential OWA weights defined by:

$$s_i = 1 - \left(\frac{i-1}{n}\right)^{0.25}, \qquad i = 1, \ldots, n \tag{13}$$

and

$$\alpha = 1/\sum_{i=1}^{n} s_i \tag{14}$$

7.3 Aggregated Ordered Achievement Functions (AOAF)

The concept of Lorenz curves outlined in Appendix A has been adapted for defining Aggregated Ordered Achievement Functions $\boldsymbol{AOAF}$.

Two types of aggregation are used which results in two $\boldsymbol{AOAF}$s denoted by $\boldsymbol{L1}$ and $\boldsymbol{L2}$, respectively. The first one represents the so-called worst conditional mean which is natural generalization of the minimum (worst) achievement aggregation. It is defined as the mean within the specified tolerance level (amount) of the worst achievements. For the simplest case one may simply define the worst conditional mean as the mean of the k worst-off achievements (or rather k/n portion of the worst achievements). This can be mathematically formalized as

$$L1_{jk} = \frac{1}{k}\sum_{l=1}^{k} OAF_{jl} \tag{15}$$

aggregating values of k-worst $OAF_j(\cdot)$). Note that for $k = 1$, $L1_{j1}$ represents the minimum achievement, and for $k = n$,

$$L1_{jn} = \frac{1}{n}\sum_{l=1}^{n} OAF_{jl} = \frac{1}{n}\sum_{i=1}^{n} AF_{ji} \tag{16}$$

which is the mean achievement.

Aggregation $L1_{jk}$ can be viewed as a simple transformation of the Absolute Lorenz Curve[19] for alternative j (denoted by ALC_j), which is defined as the PWL curve connecting the point (0,0) and points:

$$(\frac{i}{n}, \frac{1}{n}\sum_{l=1}^{i} OAF_{jl}) \qquad \text{for} \quad i = 1, \ldots, n. \tag{17}$$

Exactly,

$$L1_{jk} = \frac{n}{k} ALC_j \left(\frac{k}{n}\right). \tag{18}$$

Formula (18) is easily extendable for any (not necessarily representing an integer number) fraction of all criteria. Let this fraction be denoted by κ. Then

$$L1_j(\kappa) = \frac{1}{\kappa} ALC_j(\kappa) = \frac{1}{n\kappa}\left[\sum_{l=1}^{k} OAF_{jl} + (n\kappa - k) OAF_{j,k+1}\right] \tag{19}$$

where $k = \lfloor n\kappa \rfloor$ and the corresponding sum is equal to 0 for $k = 0$.

The second aggregation is built as the weighted sum of sorted achievements

$$L2_{jk} = \sum_{l=1}^{k-1} L2_{jl} + s_k * OAF_{jk} \tag{20}$$

with weights

$$s_k = \begin{cases} 1 & \text{if } k = 1 \text{ or } L2_{j,k-1} = 0. \\ \dfrac{1}{n * \sqrt{k}} & \text{otherwise} \end{cases} \tag{21}$$

The aggregation is similar the so-called Ordered Weighted Average (OWA) where decreasing weights s_i defined by formula (21) allow us to model decreasing importance of subsequent i/n quantiles, i.e., decreasing importance of the second worst criteria values in comparison to the importance of the worst ones, decreasing importance of the third worst criteria values in comparison to the importance of the second worst ones, etc. Although the weights defined by (21) are alternative dependent which guarantees that all the worst achievements are equally weighted but also differentiate the aggregation from the standard OWA. The relations specified above provide us with faster decreasing of importance for earlier quantiles and slower for the further ones.

8 Methods

We define here the methods currently implemented in the MCA. Each of them is defined by two functions:

- Achievement Functions $\boldsymbol{AF}$ that measures (for each criterion) the satisfaction level corresponding to a value of the criterion taking into account the relative importance of this criterion and its individual achievement.
- Scalarizing Functions $SF_j(\cdot)$ which assigns for each alternative a real value. The alternative with the largest value of SF_j is selected as the Pareto solution corresponding best to the specified preferences.

[19] See Appendix A.

8.1 Objective Choice (OC)

This is the only method which assumes equitable approach, i.e., all criteria having equal importance. It is typically used for an initial iteration, which is generated automatically, therefore user preferences are unknown (should the method be used by a user then the specified preferences are ignored).

In the OC method objective values of aspiration and reservation levels are computed first, and then used for the AFs.

8.1.1 Objective ASP/RES

Aspiration and reservation values can be defined from the values of the corresponding criterion, e.g.:

$$res_i = \alpha * aver_i \tag{22}$$

$$asp_i = \alpha * (aver_i + 1) \tag{23}$$

where the average value of i-th criterion is defined by:

$$aver_i = \sum_{j=1}^{m} q_{ij}/m \tag{24}$$

where m is the number of alternatives, and α is a given parameter (currently equal to 0.5). *Note:* values of res_i and asp_i are defined by the data, and therefore will most likely not correspond to the actual (i.e. defined by an alternative) criterion value.

8.1.2 Achievement Functions

The AFs are defined as PWL functions composed of three segments defined by the following points:
- $AF_i(nadir_i) = 0$;
- $AF_i(res_i) = 3$;
- $AF_i(asp_i) = 7$;
- $AF_i(utopia_i) = 10$;

Then the values of the $AF_i(\cdot)$ are computed as values of such PWL functions for actual values of IA_i.

8.1.3 Scalarizing Functions

The values of $\boldsymbol{AF}$ are used as arguments of $\boldsymbol{OAF}$, see eq. (10). Then SF_j are defined by:

$$SF_j = L1_j(\kappa), \qquad j \in J \tag{25}$$

where $L1_j(\kappa)$ is defined by (19), and κ is the criteria quantile that can be changed after more experiments. Currently:

$$\kappa = float(n)/3, \tag{26}$$

which implies that the selection is based on the $\boldsymbol{AOAF}$ (cumulative OAFs) defined for the worst 1/3 of criteria. Illustration of the $\boldsymbol{AOAF}$ is shown in Fig. 3.

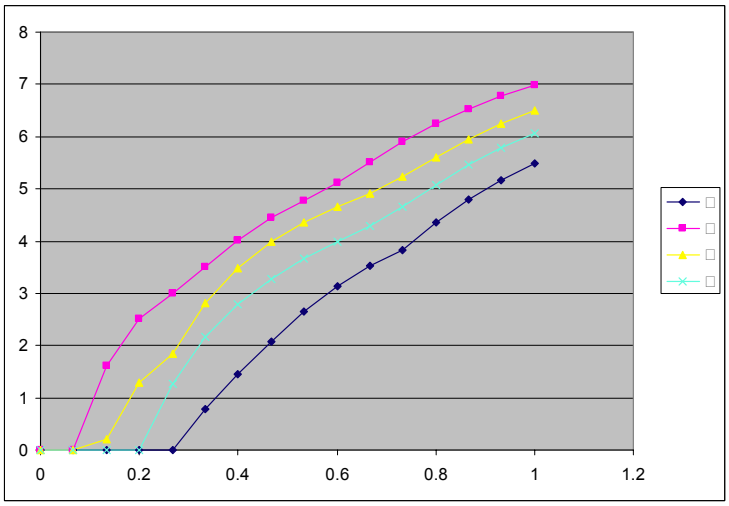

Figure 3: Examples of the Aggregated Ordered Achievement Functions $L1_j$ (15) for the problem with 4 alternatives and 16 criteria.

8.2 Aspiration-Reservation (AspRes)

The AspRes method differs from the above described OC method by replacing the computation (from the problem data) of the aspiration and reservation level values by specification of these values by the user.

The user specifies the aspiration and reservation values interactively. The Graphical User Interface (GUI) assures that aspiration values are better than the corresponding reservation values (i.e., are larger/smaller for maximized/minimized criteria, respectively). Moreover, both values are within the range defined for each criterion by the corresponding values of nadir and utopia.

8.3 RFP - Nadir

This method takes as the Reference Point (RFP) the Nadir point (defined by the worst values of all criteria), and looks for a Pareto solution along the direction defined by $\boldsymbol{w}$, see Fig. 4 for illustration.

The Scalarizing Function is defined in the following way:

- Select the improvement direction equal to the vector $\boldsymbol{w}$
- Define a ray **R** from the Nadir point in this direction
- Define line **L** by two points: j-th alternative and Nadir
- The value of SF_j is equal to the angle between the **L** and **R**.

In other words, we are looking for an alternative that improves the the criteria values at the Nadir in the proportions defined by $\boldsymbol{w}$. Such an alternative lies on the boundary of

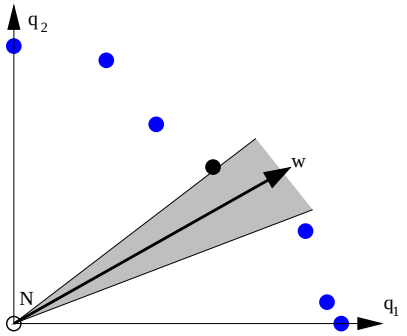

Figure 4: Nadir-based RFP method. The improvement direction is defined by the scaling/weighting vector $\boldsymbol{w}$, see eq. (3) or (9).

the cone marked in the Fig. 4 by the gray area. This approach is similar to the Weighted Sum (WS – linear criteria aggregation) approach outlined in Sec. 6.3 in the sense that criteria improvement ratios are specified as weights. However, the RFP approach avoids many deficiencies of the WS method; in particular, it supports analysis of the full Pareto set.

8.4 RFP - Utopia

This method, see Fig. 5 for illustration, is based on a concept similar to that used for the RFP-Nadir method.

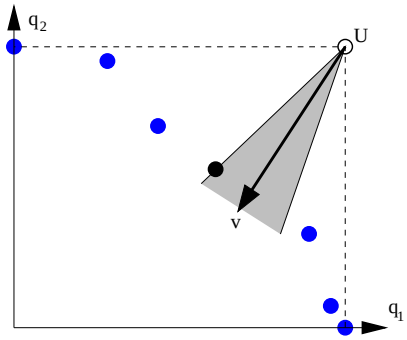

Figure 5: Utopia-based RFP method. The direction is defined by the vector $\boldsymbol{v}$, see eq. (1).

The Scalarizing Function is defined in the following way:

- Select the worsening direction equal to the vector $\boldsymbol{v}$
- Define a ray **R** from the Utopia point in this direction
- Define line **L** by two points: j-th alternative and the Utopia
- The value of SF_j is equal to the angle between the **L** and **R**.

In other words, we are looking for an alternative that worsens the the criteria values at the Utopia in the proportions defined by $\boldsymbol{v}$.

8.5 RFP - Pareto

This method finds (for a given Pareto solution denoted here as RFP) another Pareto solution that:
• has better criteria values for $i \in IMPROVE$
• compromises (if necessary) criteria values for $i \in RELAX$
• attempts to stabilize criteria values for $i \in STABILIZE$
• does not consider criteria values for $i \in FREE$

In other words, the users splits all criteria in these four disjoint sets; it is assumed that the two sets $IMPROVE$ and $RELAX \cup FREE$ must be non-empty.

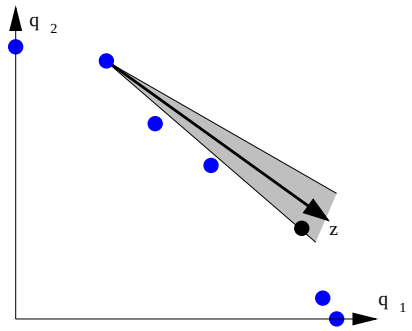

Figure 6: RFP-Pareto (not reliable) method.

We have first experimented with a method being a combination of the RFP-Nadir and RFP-Utopia methods, see Fig. 6 for illustration. For such an approach a vector $\boldsymbol{z}$ is defined as follows:

$$z_i = \begin{cases} w_i & \text{if } i \in IMPROVE \\ -v_i & \text{if } i \in RELAX \\ 0.1 * w_i & \text{if } i \in STABILIZE \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

This approach works well for some problems (especially for problems with small number of criteria), but is not satisfactory for other problems. Therefore we have implemented a simpler method, concept of which is shown in Fig. 7.

The Achievement Function is defined as:

$$AF_{ij} = sc_i * r_{ij} \tag{28}$$

where

$$r_{ij} = IA_{ij} - RFP_{ij} \tag{29}$$

$$sc_i = \begin{cases} w_i & \text{if } i \in IMPROVE \text{ and } r_{ij} > 0. \\ -\infty & \text{if } i \in IMPROVE \text{ and } r_{ij} < 0. \\ w_i & \text{if } i \in RELAX \\ 0.5 * w_i & \text{if } i \in STABILIZE \text{ and } r_{ij} > 0. \\ 2 * w_i & \text{if } i \in STABILIZE \text{ and } r_{ij} \leq 0. \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Note the following features of this method:

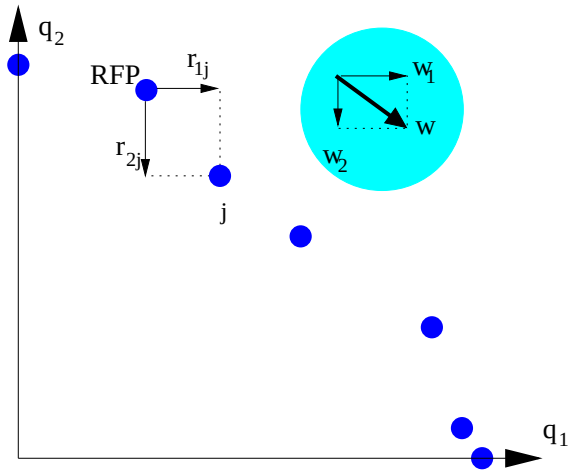

Figure 7: RFP-Pareto method.

- It never selects an alternative whose any i-th criterion (for $i \in IMPROVE$) is worse than for the RFP.
- Criteria from the set $RELAX$ that have values better than for the RFP contribute positively to the corresponding components of $\boldsymbol{AF}$ (lack of this feature is a major disadvantage of the alternative RFP-Pareto method outlined above).
- Criteria from the set $STABILIZE$ contribute to the $\boldsymbol{AF}$ much stronger/weaker depending on worse/better value for the corresponding alternative.

The Scalarizing Function is defined as:

$$SF_j = \sum_{i=1}^{n} AF_{ij} \tag{31}$$

8.6 Pairwise Outperformance Aggregation (POA)

The following dominance relation for two alternatives, say (j, l) is defined to allow for selecting (by $n-1$ pair-wise comparisons) *the best* alternative. The comparisons involve only two rows of matrix $\boldsymbol{IA}$ corresponding to the selected pair of alternatives.

- Select a weighting function $\beta(\cdot)$, e.g.:

$$\beta(x) = 10^{-x} \tag{32}$$

- Define dominance components dc as:[20]

$$di_{jlk} = w_k * (IA_{jk} - IA_{lk}) \tag{33}$$

$$dc_{jlk} = \beta(IA_{jk}) * di_{jlk} \tag{34}$$

$$dc_{ljk} = \beta(IA_{lk}) * di_{ljk} = -\beta(IA_{lk}) * di_{jlk} \tag{35}$$

[20]This definition assumes that range of $IA(\cdot)$ values is about 1. For a substantially larger/smaller ranges a scaling factor should be considered.

- Define the dominance index d as the cumulative difference of the corresponding dominance components:

$$d_{jl} = \sum_{i=1}^{n}(dc_{jli} - dc_{lji}) = \sum_{i=1}^{n}(\beta(IA_{ji}) + \beta(IA_{li})) * di_{jli} \tag{36}$$

Comments:

- One also can define a set $K = \{k : d_{jlk} > 0\}$ and interpret $k \in K$ as the sequence-numbers of criteria, at which the j-th alternative is better than the l-th alternative (in the sense of the cumulative dominance component).

Dominance definition: We say that alternative j dominates alternative l ($j \succ l$) if:

$$d_{jln} > 0 \quad \text{or} \quad \left(d_{jln} = 0 \text{ and } \sum_{k=1}^{n} di_{jlk} > 0\right) \tag{37}$$

Otherwise alternative j is (weakly) dominated by alternative l.

Comments:

- The method selects alternative that dominates all other alternatives; this is done by comparing $n-1$ pairs of alternatives.
- The difference of values of each pair defining the corresponding dominance component is weighted by:
  - $\beta(\cdot)$ in order to *amplify* the influence of weak/bad criteria values (much) more than that of good values;
  - w_i to reflect relative importance of i-th criterion.

  Note that values of dc_{jlk} and dc_{ljk} have different signs (unless both are equal to 0), and their absolute values may differ substantially, especially, if only one of the compared alternatives has the k-th criterion value that is bad.
- The dominance relation is based on a cumulative index, therefore several/many small components (contributions) can be offset by a smaller number of larger contributions.

The index $best$ of best (i.e. dominating all other) alternative is selected by the following algorithm:

- Set $b = 1$
- for(j = 2; j $\leq$ m; ++j) {
  if($j \succ b$) b = j;
  }

where the dominance relation ($j \succ b$) is defined by (37).

A more detailed description of the POA method (and 11 other methods belonging to the class of pairwise-outperformance based approaches) can be found in [2].

8.7 Non-linear Aggregation (NA)

AF are defined by:

$$AF_{ij} = w_i * (IA_{ij}(q_{ij}))^a, \qquad i \in I \tag{38}$$

where $0 < a < 1$ is the method dependent constant.

The scalarizing functions are defined by:

$$SF_j = \sum_{i=1}^{n} AF_{ij}, \qquad j \in J \tag{39}$$

The achievement functions (38) take into account two factors. First, the root of degree $1/a$ is used to amplify the influence of increasing weak/bad criteria values (much) more than that of good values. Any improvement of a given value results in a larger increase of satisfaction (utility) than the same improvement of a larger value. Experiments have shown that $a = 0.4$ works sufficiently well for most problems. Note, that this value provides at least 10 times larger impact of improvement value within the 10% worst quantile than for improvements within the 10% best quantile. Second, user defined importance weights are applied to reflect the trade-offs between satisfaction levels of the corresponding criteria.

8.8 Quantile Aggregation (QA)

This method uses two sets of weights:
- user-defined weights $\boldsymbol{w}$ which reflect relative importance of criteria,
- method-defined weights $\boldsymbol{s}$ (defined in Section 7.2) allocated to n quantiles and representing levels of aversion to criteria values (i.e., the worse the criterion value the higher the corresponding weight).

The $\boldsymbol{AF}$ is defined by

$$AF_{ji} = IA_{ji} \tag{40}$$

and the OAF by eq. (10).

The method uses the final OWA weights ω defined by:

$$\omega_{ji} = PWL(sow_{ji}) - PWL(sow_{j,i-1}), \qquad i = 1, \ldots, n; \qquad j \in J \tag{41}$$

where the PWL is defined by points (11), and $\boldsymbol{sow_j}$ by:

$$sow_{j,0} = 0; \qquad sow_{ji} = sow_{j,i-1} + ow_{ji}, \qquad i = 1, \ldots, n; \qquad j \in J \tag{42}$$

The $\boldsymbol{ow_j}$ vectors are composed of the ordered weights, i.e. is defined by $\boldsymbol{w}$ sorted in the same way as OAF defined by eq. (10), i.e. by the corresponding values of $\boldsymbol{q_j}$.

The $\boldsymbol{SF}$ is defined as

$$SF_j = \sum_{i=1}^{n} \omega_{ji} OAF_{ji}; \qquad j \in J \tag{43}$$

The aggregation (43) represents the so-called importance Weighted OWA (WOWA)[21] Usage of importance weights to intensify or abate some criteria causes that the i/n quantiles no longer correspond strictly to the worst achievement value, the second worst achievement value etc. Therefore, while referring to the latter the corresponding weights s_i have to be recalculated taking into account changes in quantiles caused by weights w_i. This is achieved by the aggregation of weights s_i through the PWL function defined by the points (11), and then their disaggregation according to the weights w_i ordered according to the corresponding achievements.

[21] See e.g., [12].

8.9 LexMaxReg

This method is an extension of the LexMax method, see e.g., [9, 11]. AF is defined by:

$$AF_{ji} = w_i * IA_{ji}(q_{ji}), \qquad i \in I \tag{44}$$

Then OAF is computed as defined by eq. (10). The LexMax method uses

$$SF_j = OAF_{kj} \tag{45}$$

for the smallest k for which the values of the k-th row of the OAF differ.

The LexMaxReg method uses the scalarizing function defined as:

$$SF_j = L2_{jn} \tag{46}$$

where $\boldsymbol{L2}$ is defined by (20).

The LexMaxReg method is similar to the LexMax (both attempt to maximize the worst criterion). However, the LexMaxReg includes in the scalarizing function (as a sort of regularizing term) all criteria. Therefore, if the differences between SF_j defined by eq. (45) are small (in comparison with other criteria) then the regularizing term included in $\boldsymbol{SF}$ defined by (46) is likely to cause selection of that has slightly worse value of the worst criterion but much better values of the other criteria.

8.10 Weighted Sum (WS)

AF are defined by:

$$AF_{ij} = w_i * IA_{ij}(q_{ij}), \qquad i \in I \tag{47}$$

and the scalarizing functions:

$$SF_j = \sum_{i=1}^{n} AF_{ij}, \qquad j \in J \tag{48}$$

This method was implemented for testing purposes. Due to its properties it is not recommended to be used for actual analysis. More detailed arguments are provided e.g., in [5], [8].

9 Comparison of selected methods

Table 1 summarizes the results of pairwise comparisons of six methods. The results are shown for each of the five selected problems; the last column contains their average. The results are the frequency of returning by both compared methods the same Pareto solution for a given preferences. Fig. 8 shows the histogram of last column in Table 1. The results of most comparisons are in the range $60 - 95\%$ (see histogram). However, there is one method $ABA1$ that appears to be an outlier: the similarity of this method to all others is below 20%.

1st method	2nd method	ch	de	fr	it	robot	average
		%	%	%	%	%	%
POA-NF	WS	66.9	70.8	63.2	78.3	46.9	66.5
POA-NF	WS-L	74.6	75.0	68.6	90.0	43.8	72.7
POA-NF	ABA1	18.2	13.5	15.7	15.0	34.4	17.2
POA-NF	NA-L	91.9	89.6	88.6	80.0	87.5	89.2
POA-NF	QA-L	92.4	89.6	78.9	93.3	62.5	86.4
OPOA-Inv-NF	WS	63.6	75.0	56.2	76.7	46.9	63.5
OPOA-Inv-NF	WS-L	75.8	81.3	73.0	85.0	43.8	75.0
OPOA-Inv-NF	ABA1	17.4	11.5	21.1	16.7	34.4	18.4
OPOA-Inv-NF	NA-L	89.4	86.5	89.2	85.0	87.5	88.3
OPOA-Inv-NF	QA-L	95.3	99.0	92.4	98.3	62.5	93.6
WS	WS-L	55.9	78.1	59.5	83.3	78.1	64.4
WS	ABA1	16.9	3.1	9.2	3.3	31.3	11.8
WS	NA-L	71.6	68.8	58.9	63.3	43.8	65.0
WS	QA-L	63.6	76.0	52.4	78.3	68.8	63.9
WS-L	ABA1	18.6	10.4	19.5	10.0	34.4	17.6
WS-L	NA-L	70.8	69.8	70.3	70.0	43.8	69.0
WS-L	QA-L	78.4	82.3	72.4	83.3	62.5	76.8
ABA1	NA-L	16.9	14.6	17.3	26.7	34.4	18.6
ABA1	QA-L	17.4	11.5	22.2	16.7	34.4	18.7
NA-L	QA-L	88.6	85.4	85.4	83.3	56.3	84.9

Table 1: Pairwise comparisons of methods.

10 Ranking of alternatives

In order to avoid the *rank reversal problem* ranking of alternatives is done by solving sequentially the following problems:
1. Start with the full set of alternatives defined for the problem instance.
2. Select the Pareto solution[22] fitting best the selected preferences.
3. Remove the selected solution from the set of alternatives, and go to step 2.

11 Other functions of the solvers

11.1 Preprocessing

1. Mark dominated (not Pareto-optimal) alternatives.
2. Prepare basic statistics on criteria and the above defined functions (distribution of values, quantiles, median).
3. Precompute elements needed for fast composition of criteria charts.
4. Select intermediate-level criteria for which no preferences should be defined (because they will have to be ignored). Such criteria are:

[22]Note that more than one solution can be selected.

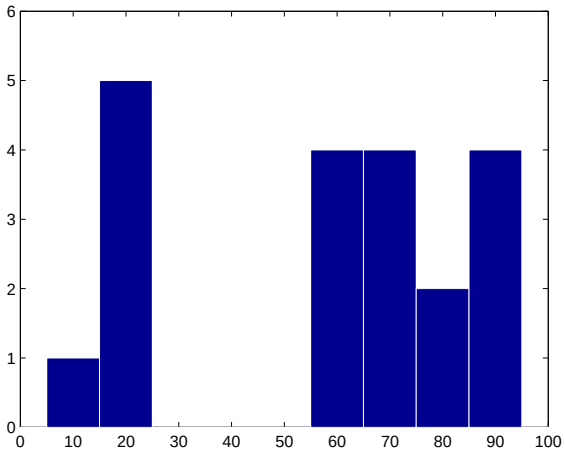

Figure 8: Histogram of the average frequency of the comparison results.

- if a higher-level criterion is the only child;
- if the leaf-criterion is the only child, then preferences are not specified for its parent.

5. Define the structures needed for possibly fast execution of the GUI (run as a Web client).

11.2 Info provided with *the best* alternative

Generally interaction should be done in the criteria space, i.e., we assume that the users will consider/analyze trade-offs in terms of criteria values. The *best* alternative (i.e., the one corresponding best to the specified preferences) implies the corresponding trade-offs between criteria values.

1. For facilitating interaction (aimed at verifying/modifying preferences) we provide possibly clear information about the criteria of *best* alternative, and help in examining feasible changes in the criteria space.
2. The user can display for any selected subset of alternatives the characteristics of:
   - criteria for which the alternative is *strong*
   - criteria for which the alternative is *weak*
3. Ranking resulting from the procedure described in Sec. 10 is provided in the graphical form.
4. Values of the $\boldsymbol{SF}$ are also available as a chart.

11.3 More info about the MCA

Detailed user guide and tutorial to the MCA is available in [6]. It contains also information about access to the MCA, which is free for research and educational purposes.

References

[1] GRANAT, J., AND MAKOWSKI, M. Multicriteria methodology for the NEEDS project. Interim Report IR-09-10, International Institute for Applied Systems Analysis, Laxenburg, Austria, 2009.

[2] GRANAT, J., MAKOWSKI, M., AND OGRYCZAK, W. Multiple criteria analysis of discrete alternatives with a simple preference specification: Pairwise-outperformance approaches. Interim Report IR-09-23, International Institute for Applied Systems Analysis, Laxenburg, Austria, 2009.

[3] LOOTSMA, F. *Multi Criteria Decision Analysis via Ratio and Difference Judgement*, vol. 29 of *Applied Optimization*. Kluwer Academic Publishers, Boston, London, 1999.

[4] LORENZ, M. Methods of measuring the concentration of wealth. *Publications of the American Statistical Association 9* (1905), 209–219. doi:10.2307/2276207.

[5] MAKOWSKI, M. Management of attainable tradeoffs between conflicting goals. *Journal of Computers 4*, 10 (2009), 1033–1042. ISSN 1796-203X.

[6] MAKOWSKI, M., GRANAT, J., AND REN, H. User guide to MCA: Multi-criteria analysis of discrete alternatives with a simple preference specification. Interim Report IR-09-22, International Institute for Applied Systems Analysis, Laxenburg, Austria, 2009.

[7] MAKOWSKI, M., GRANAT, J., REN, H., SCHENLER, W., AND HIRSCHBERG, S. Requirement analysis and implementation of the multicriteria analysis in the NEEDS project. Interim Report IR-09-09, International Institute for Applied Systems Analysis, Laxenburg, Austria, 2009.

[8] MAKOWSKI, M., AND WIERZBICKI, A. Modeling knowledge: Model-based decision support and soft computations. In *Applied Decision Support with Soft Computing*, X. Yu and J. Kacprzyk, Eds., 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`.

[9] OGRYCZAK, W. On the lexicographic minimax approach to location problems. *European Journal of Operational Research 100* (1997), 566–585.

[10] OGRYCZAK, W., AND SLIWINSKI, T. On equitable approaches to resource allocation problems: The conditional minimax solutions. *Journal of Telecommunications and Information Technology*, 3 (2002), 40–48.

[11] OGRYCZAK, W., AND SLIWINSKI, T. On direct methods for lexicographic min-max optimization. In *Computational Science and Its Applications-ICCSA 2006*, no. 3982 in Lecture Notes in Computer Science. Springer, Berlin, 2006, pp. 802–811.

[12] TORRA, V. The weighted OWA operator. *Int. J. Intell. Syst. 12* (1997), 153–166.

[13] WIERZBICKI, A., MAKOWSKI, M., AND WESSELS, J., Eds. *Model-Based Decision Support Methodology with Environmental Applications*. Series: Mathematical Modeling and Applications. Kluwer Academic Publishers, Dordrecht, 2000. ISBN 0-7923-6327-2.

[14] YAGER, R. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Trans. Systems, Man and Cyber. 18* (1988), 183–190.

[15] YAGER, R., AND KACPRZYK, J., Eds. *The Ordered Weighted Averaging Operators: Theory and Applications*. Kluwer, Dordrecht, 1997.

A Lorenz curve and quantile measures

The concept of Lorenz curves is over 100 years old, see [4]. In income economics the Lorenz curve is a cumulative population versus income curve. A modification of the classical example is shown in Fig. 9. It represents cumulative distribution of several selected levels of household incomes (sorted by increasing levels), e.g., about 30% of the bottom households accumulates about 5% of total income. Lorenz curve is (weakly) increasing when built for positive (nonnegative) outcomes. It is always convex (provided that the mean value is positive), but not necessarily strictly convex. A perfectly equal income distribution would result in equal income for each household. The corresponding Lorenz curve would be a line with slope of 45 degree. Another extreme is that one household has all the income. Then the Lorenz curve would be composed of a flat line and a single point.

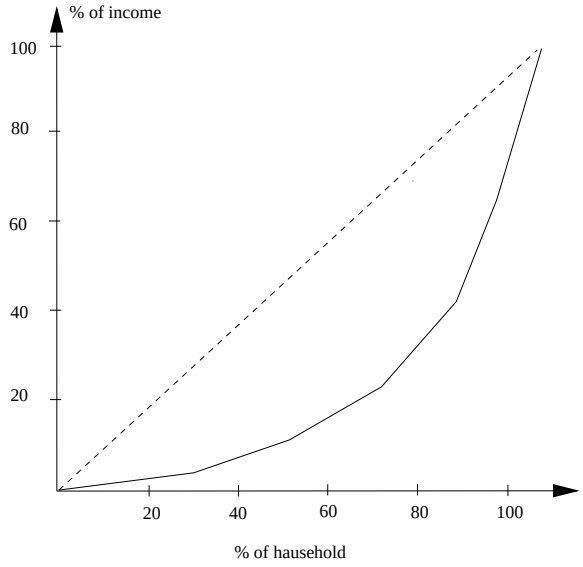

Figure 9: Lorenz curve illustrating distribution of household incomes.

A similar approach can be used for illustrating the cumulative distribution of the $\boldsymbol{OAF}$ values. Although the original Lorenz curve is focused on equity while ignoring the mean result. Recall that any perfectly equal distribution of income as the Lorenz curve has the diagonal line (the same independently from the income value) and no other income vector can be better. Therefore, in the so-called equitable optimization the Absolute Lorenz Curves (ALC) are used. The ALCs are not normalized, i.e., they take into account also values of the achievements. An ALC is defined as the PWL curve connecting point (0,0) and points

$$(\frac{i}{n}, \frac{1}{n}\sum_{l=1}^{i} OAF_{lj}), \qquad i = 1, \ldots, n$$

thus ending at the point $(1, \mu_j)$, where

$$\mu_j = \frac{1}{n}\sum_{l=1}^{n} AF_{jl}$$

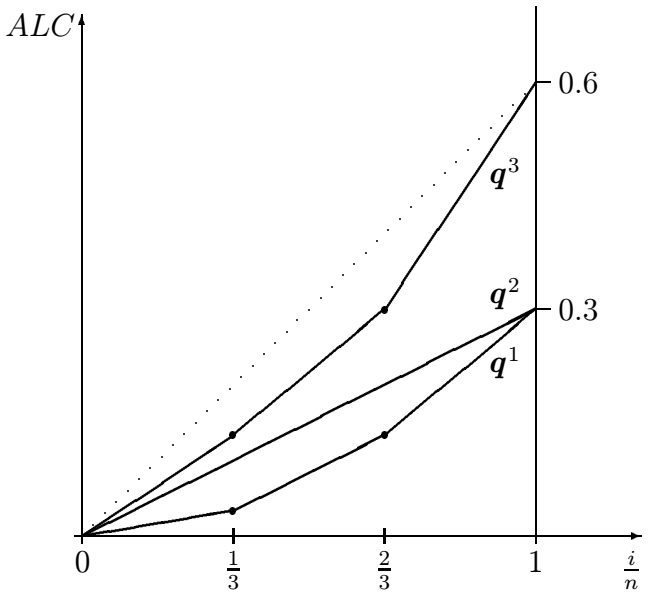

Figure 10: Absolute Lorenz Curves.

represent the mean (average) achievement for given alternative. The original Lorenz curve is defined by points

$$(\frac{i}{n}, \frac{1}{n\mu_j} \sum_{l=1}^{i} OAF_{lj}), \qquad i = 1, \ldots, n$$

and therefore it ends at point $(1, 1)$.

Within the ALC model vectors of equal achievements are further distinguished according to their achievement values. The achievements are graphically represented with various rising lines. Vector of equal achievements obviously dominates any unequal vector with the same mean. However, a vector of unequal but larger achievements may be preferred to a vector with smaller although equal achievements.

Fig. 10 presents the ALCs for three achievement vectors. One can easily see that vector of perfectly equal achievements $\boldsymbol{q}^2 = (0.3, 0.3, 0.3)$ dominates vector $\boldsymbol{q}^1 = (0.1, 0.3, 0.5)$ having the same mean value. However, $\boldsymbol{q}^2$ is dominated by $\boldsymbol{q}^3 = (0.4, 0.5, 0.9)$ composed of unequal achievements with larger ordered values.

For the four alternatives defined in the problem presented in Sec. 8.1.1 the ALCs are shown in Fig. 11. Note that in this example curves do not cross. This is not always the case. For other problems alternatives have more diversified distributions of criteria values. Therefore the corresponding Lorenz curves are flat for several criteria (for which the values are bad), and then increase rapidly for the other criteria. Depending on the pattern of criteria importance set by the user such alternatives might be strongly preferred or strongly disliked.

Lorenz curves may provide experienced users with useful characteristics of the alternatives. However, they can be misleading for less experienced users. Therefore we have decided to not display them. Although they are used to build some aggregations.

The so-called worst conditional[23] mean is a natural generalization of the minimum (worst) achievement aggregation, see Fig. 12 for illustration, and [10] for details. It is

[23]The *conditional* here means that one considers only values worse than the corresponding threshold.

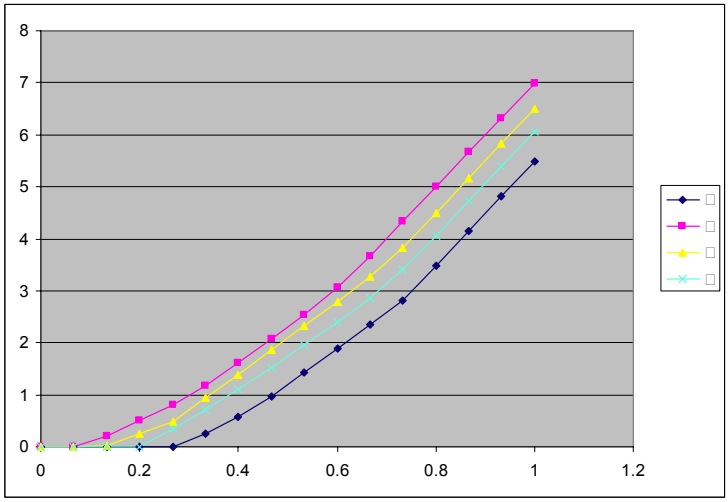

Figure 11: Lorenz curve for the alternatives presented in Sec. 8.1.1.

defined as the mean worst achievements within the specified threshold level. For the simplest case one may simply define the worst conditional mean as the mean for the k worst achievements (or rather k/n fraction of the worst achievements). This can be mathematically formalized as

$$\mu_j(\frac{k}{n}) = \frac{1}{k}\sum_{l=1}^{k} OAF_{jl} = \frac{n}{k} ALC_j(\frac{k}{n}) \tag{49}$$

aggregating values of k-worst OAF_{jl}. Note that for $k = 1$, $\mu_j(1/n)$ represents the minimum achievement, and for $k = n$, it reaches the mean achievement $\mu_j(n/n) = \mu_j$. Relation to the ALC_j extends the definition for any (i.e., not necessarily representing an integer number of criteria) fraction of criteria. If the latter is denoted by κ, then

$$\mu_j(\kappa) = \frac{1}{\kappa} ALC_j(\kappa). \tag{50}$$

The worst conditional mean actually takes into account only single points of the ALC. Now we will discuss other aggregations that take into account entire curves. The ALC may be also used for illustrating the concept of the Ordered Weighted Average (OWA), see e.g., [14, 15].

$$\sum_{i=1}^{n} s_i OAF_{ji} \tag{51}$$

where decreasing quantile weights s_i are assigned to the ordered achievements. The quantile weights s_i are used for representing decreasing importance of subsequent i/n quantiles, i.e., the decreasing importance of the second worst criteria values in comparison to the importance of the worst ones, then the further decreasing importance of the third worst criteria values in comparison to the importance of the second worst ones, etc. Indeed

$$\sum_{i=1}^{n} s_i OAF_{ji} = \sum_{i=1}^{n} \bar{s}_i ALC_j(\frac{i}{n}) \tag{52}$$

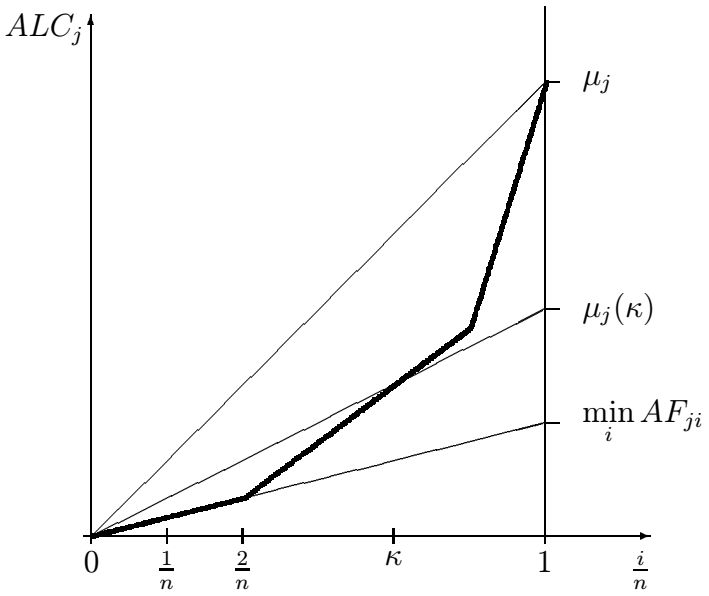

Figure 12: Absolute Lorenz Curve and the worst conditional mean.

with weights

$$\bar{s}_i = s_i - s_{i+1}, \qquad i = 1, \ldots, n-1; \qquad \bar{s}_n = s_n. \tag{53}$$

Hence, the OWA aggregation with the decreasing quantile weights may be viewed as the weighted arithmetic mean of the ALC segments with appropriate positive weights.

For the curves corresponding to the four alternatives illustrated in Fig. 11 the alternative B is dominating all other alternatives in the sense that it has the most equal distribution of criteria values as well as the largest worst conditional means. For equitable preferences (all criteria has the same importance) the alternative B might be also preferred by the user. However, this reasoning will not be justified if the importance the user attaches to criteria differ. A justification for selecting another alternative might be that the criteria with higher importance have better values (note that the OAFs are sorted, and typically different criteria are the worst ones for different alternatives).

The OWA aggregation (51) is built for equally important achievements where only the distribution of achievements values is evaluated. Actually, achievement vectors having the same OWA value may differ only by the order of achievement values. For instance, consider two symmetric achievements vectors $\boldsymbol{q}^1 = (0, 1)$ and $\boldsymbol{q}^2 = (1, 0)$, and OWA weights $s_1 = 0.9$ and $s_2 = 0.1$; the OWA aggregation for both vectors is equal:

$$OWA_1 = OWA_2 = 0.9 \cdot 0 + 0.1 \cdot 1 = 0.1.$$

However, the users typically want to associate different importance to the elements of achievement vectors. This can be done by introducing into the OWA aggregation the importance weights w_i, which define a repetition measure within the distribution (population) of achievement values. Note, that the OWA weights s_i are applied to the averages within specific quantiles of size $1/n$ for this distribution. To illustrate this concept let us consider the importance weights $w_1 = 0.75$ and $w_2 = 0.25$. Then the achievement vector $\boldsymbol{q}^1 = (0, 1)$ is represented by the distribution having the value 0 with the repetition measure 0.75, and the value 1 with the repetition measure 0.25; the achievement vector $\boldsymbol{q}^2 = (1, 0)$ is represented by the distribution having the value 1 with the repetition

measure 0.75, and the value 0 with the repetition measure 0.25. In this specific case, the distributions may be equivalently interpreted in four dimensional space of equally important achievements (applying the measure of 0.25 to each element) where the original first achievement has been triplicated; thus $\bar{\boldsymbol{q}}^1 = (0, 0, 0, 1)$ and $\bar{\boldsymbol{q}}^2 = (1, 1, 1, 0)$. The OWA aggregation with weights $s_1 = 0.9$ and $s_2 = 0.1$ applied to the corresponding averages within quantiles of size 0.5 results in the aggregation values $0.9 \cdot 0 + 0.1 \cdot (0 + 1)/2 = 0.05$ for $\bar{\boldsymbol{q}}^1$, and $0.9 \cdot (0 + 1)/2 + 0.1 \cdot 1 = 0.55$ for $\bar{\boldsymbol{q}}^2$, respectively.

Certainly, one do not need to transform all the cases to equally important achievements in order to calculate the appropriate OWA value. The WOWA aggregation[24] provides a way for including the importance weighting into the OWA concept. The WOWA is defined as:

$$WOWA_j = \sum_{i=1}^{n} \omega_i OAF_{ji} \quad \text{with} \quad \omega_i = PWL(\sum_{k=1}^{i} w_{\tau(k)}) - PWL(\sum_{k=1}^{i-1} w_{\tau(k)}) \qquad (54)$$

where PWL denotes the piecewise linear function representing the linear interpolation of the points

$$(\frac{i}{n}, \sum_{k=1}^{i} s_k)$$

together with the point $(0, 0)$, and τ represents the ordering permutation for AF_{ji} (i.e., $AF_{j\tau(i)} = OAF_{ji}$).

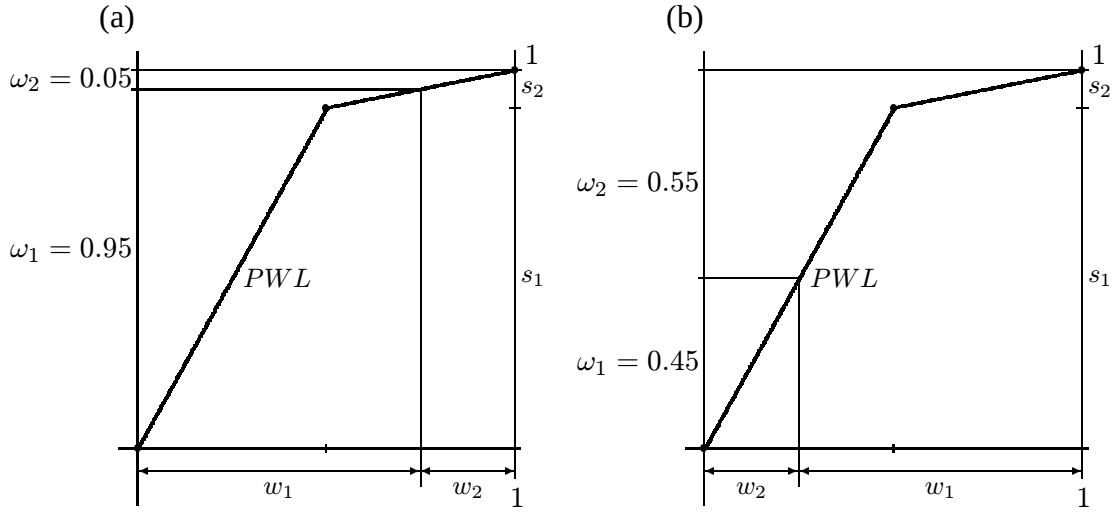

Figure 13: WOWA calculation according to formula (54).

Fig 13 illustrates application of formula (54) to the example discussed above, i.e., two achievement vectors $\boldsymbol{q}^1 = (0, 1)$ and $\boldsymbol{q}^2 = (1, 0)$, OWA weights $s_1 = 0.9$ $s_2 = 0.1$, and the importance (reflecting the user preferences) weights $w_1 = 0.75$, $w_2 = 0.25$. Fig 13 (a) illustrates the calculation of the $\omega = (0.95, 0.05)$ for the achievement vector $\boldsymbol{q^1}$, which results in $WOWA^1 = 0.95 \cdot 0 + 0.05 \cdot 1 = 0.05$. Fig 13 (b) illustrates the calculation of the ω for $\boldsymbol{q^2}$, which results in $WOWA^2 = 0.45 \cdot 0 + 0.55 \cdot 1 = 0.55$, thus showing that $\boldsymbol{q^2}$ clearly dominates $\boldsymbol{q^1}$.

[24] See e.g., [12].