SoftwareX 27 (2024) 101801

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original Software Publication

pyMCMA: Uniformly distributed Pareto-front representation

Marek Makowski [a,b,*], Janusz Granat [c], Andrii Shekhovtsov [c], Zbigniew Nahorski [a], Jinyang Zhao [d]

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

ARTICLE INFO

Keywords:
Pareto-front representation
Multiple-criteria model analysis
Pareto-front visualization
Pyomo modeling language
Structured modeling

ABSTRACT

`pyMCMA` is the Python implementation of a novel method for autonomous computation of the Pareto-front representation composed of efficient solutions distributed uniformly in terms of distances between neighbor Pareto solutions. `pyMCMA` supports scientific, i.e. objective, model analysis by providing preference-free Pareto front representation. `pyMCMA` seamlessly integrates independently developed substantive models. The computed Pareto-front, also for more than two criteria, is visualized by interactive parallel coordinate plot, as well as by charts of criteria pairs. Moreover, `pyMCMA` optionally exports the results for problems-specific analysis in the substantive model's variables space. The `pyMCMA` functionality is illustrated by an analysis of China's liquid fuel production model.

Code metadata

Current code version	1.1.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-24-00145
Permanent link to Reproducible Capsule	Not applicable
Legal Code License	GNU General Public License (GPL)
Code versioning system used	git
Software code languages, tools, and services used	Python 3.11, conda 24.1
Compilation requirements, operating environments & dependencies	macOS or Linux or MS-Windows. All needed packages are installed at the `pyMCMA` installation
If available Link to developer documentation/manual	https://pymcma.readthedocs.io
Support email for questions	marek@iiasa.ac.at

1. Motivation and significance

Development of the `pyMCMA` software is motivated by the authors' extensive experience in research on the Multiple-Criteria Model Analysis (MCMA) methods and tools, as well as their applications to modeling diverse complex problems. This experience has shown the limitations of the known approaches which hinder the research effectiveness of model-based problem analysis and solving.

Table 1 summarizes the basic terms necessary for understanding the methodology applied in the presented software. Extensive discussions of the MCMA methodology can be found e.g., in [1–8], and references therein.

1.1. Contributions to scientific discovery

Contributions of MCMA in scientific discovery are similar to those of optimization theory and the corresponding solvers. None of them solves any research question directly. However, the scientific discovery would be hardly possible without the application of optimization in general, and of the MCMA in particular. `pyMCMA` enhances the MCMA capabilities by:

- autonomous and effective computation of the uniformly distributed (in the criteria space) Pareto front representation;
- new method for sequential improvement of the representation, see Section 1.2.1;

* Corresponding author at: Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.
E-mail address: marek@iiasa.ac.at (Marek Makowski).

https://doi.org/10.1016/j.softx.2024.101801
Received 7 March 2024; Received in revised form 18 May 2024; Accepted 12 June 2024

Table 1
Summary of the not-commonly known terms used in the paper.

Short	Term	Definition	Notes
–	Core model	Mathematical programming model of the analyzed problem; a.k.a. substantive model.	A core model has to be provided for any MCMA tool.
–	Pareto solution	A solution of multiple-criteria problem such that no other solution exists with at least equally good values of all criteria.	In pyMCMA: ϵ-properly Pareto solution, see e.g. [6].
–	Pareto front	Set of all Pareto solutions; a.k.a., Pareto set, set of efficient solutions, set of non-dominated solutions.	Pareto solutions are objectively incomparable.
U, N	Utopia, Nadir	Points defined by the vectors of the criteria values, the best and worst within all Pareto solutions, respectively.	Hyper-cube defined by normalized U, N contains all Pareto solutions.
RFP	Reference Point	Methodology for exploring Pareto front.	Commonly used methodology.
A, R	Aspiration, Reservation	Values of each criterion that the user wants to achieve and avoid, respectively.	Also A/R-based specification of preferences [9,10]
CAF	Criterion Achievement Function	A strictly monotonic function of the corresponding criterion, decreasing or increasing for minimized or maximized criteria, respectively. CAF normalizes achievements of criteria to the common scale [0, 100].	pyMCMA applies the CAFs parameterized by the A/R values, see, e.g., [10].
AF	Achievement Scalarizing Function	A function aggregating the CAFs of all criteria. Maximization of AF provides ϵ-properly efficient Pareto solution fitting best the A/R preferences.	AF used in pyMCMA is defined in [10], and outlined in Section 1.2.1.
–	Cuboid	In pyMCMA: (hyper-)cuboid defined by two neighbor Pareto solutions.	See Section 1.2.1

- visualization of the Pareto front also for more than three criteria cases;
- seamless integration of the core model with pyMCMA;
- effective support for in-depth analysis in the core-model variables' space.

1.2. Research problem addressed

MCMA involves consideration of reaching diverse and mutually either conflicting or (partly) synergistic (consensual, concordant) objectives, such as costs, quality, negative and positive impacts, risks, ranges of desired outputs, etc. Thus, each solution is characterized in the criteria space by the vector of the criteria values. All MCMA methods focus on the Pareto front composed of Pareto solutions, i.e., such solutions of multiple-criteria problem that no other solution exists with at least equally good values of all criteria. Each Pareto solution is preferred over a subset of non-Pareto (dominated by that solution) feasible core-model solutions.

Pareto solutions are objectively incomparable and the solution choice depends on subjective preferences for trade-offs between simultaneously attainable goals for often conflicting objectives. Prevailing MCMA approaches assume the preferences defined by each user. The selection of Pareto solutions fitting best the specified preferences is well-researched and supported by diverse methods of preference specification, and by the corresponding software, typically enabling interactive analysis. Such analysis focuses on a small Pareto-front subset of solutions fitting best the selected preferences.

However, the above-summarized approach poorly supports the scientific problem analysis that has to be objective. MCMA, in this context, needs to equitably treat all objectives. To be objective, a scientific analysis should be preference-free and uniformly cover the whole set of Pareto solutions. Thus, a uniformly distributed Pareto-set representation is strongly desired. Yet, experience from analysis of diverse actual models summarized in Section 4 shows, that it is difficult and time-consuming to obtain such a representation interactively for problems with more than two criteria.

The paper addresses the need for scientific analysis by presenting a novel methodology and the corresponding open-source software. The developed method assures autonomous (without asking the user to provide any parameter characterizing the model) computation of the Pareto-front representation composed of efficient solutions that are uniformly distributed in terms of distances between neighbor Pareto solutions. The method efficiently finds, at every iteration, the most distant neighbor solutions, and then computes a new Pareto solution between them, thus sequentially improving the Pareto-front representation. The developed pyMCMA Python package enables seamless integration of the MCMA with independently developed substantive models. The computed Pareto-front, also for more than three criteria, is visualized by interactive parallel-coordinates plot, as well as by charts of the criteria pairs. Moreover, pyMCMA optionally exports the results for problems-specific analysis in the substantive model's variables space. The pyMCMA functionality is illustrated by MCMA of China's liquid fuel production model.

Therefore, pyMCMA fills the gap in the commonly known MCMA methods by providing the scientific, i.e. objective, multiple-criteria model analysis with a new methodology and its implementation in an easy-to-use software tool. The software is available under the GNU General Public License (GLP).

1.2.1. Methodology developed for pyMCMA

The methodology developed for pyMCMA originated from the successful A/R methodology of the sequential calculation of Pareto solutions based on preference specification in terms of the criteria aspiration and reservation values, see [10]. In the approach presented in this paper, instead of the user-specified A/R, they are defined by the achievements of the selected, previously computed Pareto solutions.

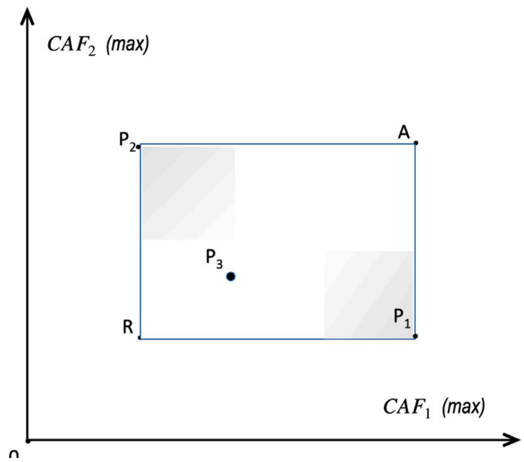

Fig. 1. Illustration of cuboid defined by Pareto solutions P_1 and P_2.

The presented approach is based on computing the Pareto front representation through sequentially generated cuboids defined by pairs of neighbor Pareto solutions. Then, each cuboid is used to define the A/R values. Next, the newly calculated Pareto solution fills the gap in the so-far computed Pareto-front representation. The procedure is repeated until the assumed distribution of the representation is reached.

Fig. 1 illustrates the cuboid construction for bi-criteria problems. Let N be a set of indices (i, j) of neighbor Pareto solutions pairs, and K be a set of criteria indices k. Next, find the most distant neighbor solutions:

$$(P_1, P_2) = \arg \max_{(i,j)\in N} \|P_i - P_j\|_{L^\infty} \tag{1}$$

Note, that each point in (P_1, P_2) is defined by the corresponding vector of the Criterion Achievement Function (CAF) values, i.e.,

$$P_i = (vcaf_{ik}), \quad k \in K, \ i \in N, \tag{2}$$

where $vcaf_{ik}$ is the value of CAF for kth criterion in ith solution. The CAFs are defined as the Piece-Wise Linear (PWL) functions of the core-model variable q_k representing the corresponding criterion. The CAF for kth criterion is denoted by $caf_k(\cdot)$, and it is parameterized by the A/R values in such a way that it is strictly decreasing or increasing, for minimized and maximized criteria, respectively. The details of the CAF specifications are available in [10].

Next, define the A/R values for each criterion by the corresponding criteria achievements of the neighbor solutions:

$$a_k = \max(vcaf_{1k}, vcaf_{2k}), \ k \in K, \tag{3}$$

$$r_k = \min(vcaf_{1k}, vcaf_{2k}), \ k \in K. \tag{4}$$

Note that the `max` and `min` operators are applied for selecting (from two solutions defining the cuboid) for each criterion the better and the worse achievement, respectively. Note that the better/worse corresponds to the larger/smaller CAF value, and thus is independent of the criterion (max- or minimized) type.

Next, define the Achievement Function $AF(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{r})$, and the optimization problem providing Pareto solution $P(\boldsymbol{q})$:

$$P(\boldsymbol{q}) \ = \ \arg\max_{x\in X} \left(AF = \min_{k\in K} caf_k(q_k, \ a_k, \ r_k) + \frac{\epsilon}{n} \cdot \sum_{k\in K} caf_k(\cdot)\right) \tag{5}$$

where $\boldsymbol{q}$ stands for the core-model variables representing the corresponding criteria, $\boldsymbol{a}, \boldsymbol{r}$ are the vectors of the A/R values, respectively, ϵ is a small positive parameter assuring computation of an ϵ-properly Pareto solution (see [6] for the methodology explanation), n denotes the number of criteria, and X is the set of feasible solutions.

Thanks to the properties of the A/R method, maximization of (5) provides the ϵ-properly Pareto solution best fitting the A/R preferences, see [6,10] for details. Moreover, because the A/R are defined by the neighbor Pareto-solutions, $P(\boldsymbol{q})$ will be within the above-defined cuboid unless (P_1, P_2) define a gap in the Pareto set. The new solution is marked in Fig. 1 by P_3 and will be considered, together with all previously computed solutions, for defining new cuboids.

Additionally, the approach guarantees that every new Pareto solution improves, in terms of uniform distribution of criteria values, the Pareto-front representation. This property is implied by the location of each new solution inside the corresponding, sequentially generated, cuboid having vertices defined by the pair of the most distant Pareto solutions.

The algorithm implementing the above-outlined method cannot be presented here due to the constrained article length. However, some implementation details are summarized in Section 2.1.

1.3. How the software is used

`pyMCMA` requires, as any MCMA tool, a prior development of the core model, which should conform to commonly accepted good modeling practice, as well as to obvious requirements on the specification of variables to be used as criteria. The model can be developed on another computer. The online documentation includes the recommendation of the model preparation. Moreover, `pyMCMA` distribution includes an example of a tiny core model.

Typically, each core model is subject to many analyses, involving e.g., diverse combinations of criteria. Therefore, `pyMCMA` supports the organization of diverse analyses by definition of each analysis in a simple configuration file located in the corresponding dedicated directory, where also the analysis results are provided.

`pyMCMA` is run by simple terminal commands. The online documentation discusses all elements of the `pyMCMA` use. Moreover, `pyMCMA` installation includes its testing, which also serves as a demonstration of the use.

1.4. Related research

We discuss the MCMA only, i.e., refrain from discussing the MCDA (Multi-Criteria Decision Analysis) dealing with the analysis of a given set of discrete alternatives.

The methodology developed for `pyMCMA` is based on the paradigm described in [10], which is not similar to any of the various methods developed for the Pareto-front representations. To the best of the authors' knowledge, `pyMCMA` is the first implementation of autonomous, parameterization-free computation of uniformly distributed Pareto-front representation. We also point out that many published methods are suitable only for problems with less than four criteria.

Although the presented method substantially differs from all known approaches, it builds on a huge legacy of methods developed over 70 years. In a very short article focused on the software description, it is impossible to provide a survey of the methods, which either directly or indirectly contributed to the `pyMCMA` development. Also, a discussion of numerous and diverse approaches to Pareto-front generation or examination of Pareto-front subsets is beyond the scope of this paper.

Therefore, the best one can do to acknowledge the MCMA legacy is to cite a sample of several dozens of publications directly related to the Pareto-front representation, augmented by the much smaller sample of general MCMA-related publications. The selection of citations in Table 2 is biased towards the publications related, in various ways, to the `pyMCMA` development. Finally, note that grouping of the citations in Table 2 is somewhat arbitrary, as most of the publications address several topics.

Table 2
References to the related publications.

Theories and methods	[2–4,11–15]
Reference point approaches	In countless publications
Aspiration/reservation methods	[16] [9] [6,10,17].
Diverse methods for Pareto-front representations	[18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31]
Methods for generation or examination of parts of Pareto front conforming to diverse specifications	[32] [33] [34] [35] [36] [37] [38] [39] [40]
Measuring quality of Pareto set approximation	[41] [42]
Methods of Pareto-front pruning and/or clustering	[43] [44] [45] [46] [47]
Domain-specific Pareto-set representations	[48] [49] [50]
Finding Pareto-front knees	[51] [52] [53]
Visualization	[54] [55]

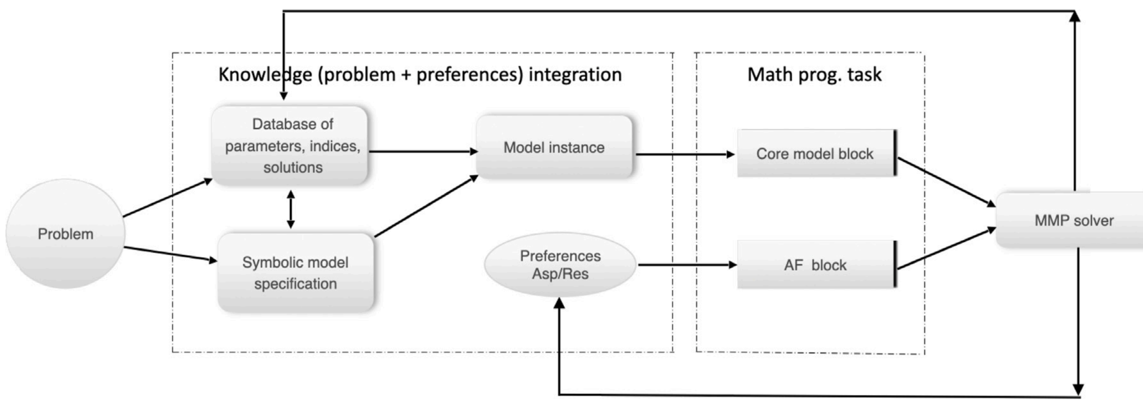

Fig. 2. Architecture of pyMCMA.

2. Software description

2.1. Software architecture

We outline the pyMCMA software architecture although it is transparent for the pyMCMA users, who use it in a similar way as optimization solvers, i.e., typically without knowledge of the software implementation.

In the MCMA context, the mathematical programming problem of Pareto-solution selection consists of: (1) the core model, and (2) a model representing preferences for Pareto-solution selection. Section 1.3 summarizes the core-model requirements, and Section 1.2.1 presents methodology applied in pyMCMA for autonomous, sequential specification of preferences leading to generation of uniformly distributed Pareto-front representation.

Fig. 2 illustrates the architecture of pyMCMA, implemented in Pyomo, the Python-based, open-source optimization modeling language [56]. Pyomo supports the structured modeling paradigm, see e.g., [57], by separating the symbolic model specification (called the abstract model), and a model instance (called the concrete model). Moreover, Pyomo enables seamless integration of concrete models by creating a new concrete model composed of blocks, where each block is defined by one of the integrated models. Therefore, pyMCMA uses the provided core model as one block and sequentially generates the second block representing the model outlined in Section 1.2.1. The latter model is marked in Fig. 2 as the *AF block*, and further on referred to as the *AF-model*.

In terms of the mathematical programming, the AF defined in (5) is implemented as an abstract Linear Programming (LP) model. Elements of the Pareto-front representation are generated in the iterative process. In each iteration, the abstract model is instantiated by the A/R values defined by (3) and (4), and seamlessly integrated with the core model using the Pyomo blocks concept. The resulting concrete model is then solved by the Pyomo solver.

Note that each instance of the *AF-model* is as a small LP problem. Therefore, the type of the resulting optimization problem is determined by the class of the core model. Thus, the computational requirements are similar to that of a single-criterion optimization of the core model. The current pyMCMA version is solved by the standard Pyomo solver GLPK, and it was tested with the LP core model.

Solutions of the integrated model sequentially generated using the corresponding cuboids, are used in two ways:

- By pyMCMA for Pareto-front representation, marked by the lower path.
- Optionally, for the core-model post-MCMA dedicated analysis, marked by the upper path. This option is explained in the online documentation.

The *AF-model* is generated in two steps. First, the abstract model implements the relations discussed in Section 1.2.1. This part is invariant during the computations. Second, the solutions defining cuboids are selected, and the corresponding A/R values provide the data for instantiating the *AF-model*.

2.2. Software functionalities

2.2.1. Functional features

pyMCMA extends the commonly known MCMA functionality by: (1) seamless integration of diverse, independently developed, core models with the MCMA tool, (2) supporting customized computations of the Pareto front representations, and (3) providing configurable data for the problem-specific analysis in the core-model decision and state variables space. A summary of the functionality pertaining to each of these issues follows.

1. Integration of the MCMA tool with the core-models:

- Development of core models can be done independently of their analysis, often on different hardware and/or software platforms. The core model, after testing, is exported in a portable format supported by the standard Python package. The online documentation provides the details and the software distribution includes the corresponding example of the core-model export.
- pyMCMA imports the core model provided in the portable format, and seamlessly integrates it with the *AF-model*.

Therefore, pyMCMA follows the proven way to support good modeling practice and increases the efficiency by providing modular, platform-independent, and reusable software tool.

2. Computation of the Pareto-front representation consists of autonomously generated iterations. It results in: (1) generation of Pareto-efficient solutions, and (2) visualization of the generated solutions. Summing up:

    1. Solutions included in the Pareto-front representation have the values of $CAF(\cdot)$ as uniformly distributed as possible. In other words, the implementation guarantees that the maximum distance between neighbor solutions is minimized but the solutions closer than the defined resolution of the representation are excluded. The L^{∞} norm is applied as the distance measure to assure consistency with the norms used in underlying A/R method [10], which in turn was motivated by the Principles of Justice [58]. Moreover, in cases of disconnected Pareto sets, the corresponding gaps do not influence the minimization of distances between other solutions. Section 1.2.1 provides details on the underlying methodology of the implemented representation.
    2. The computed Pareto-front representation is visualized to support initial analysis in the criteria space. pyMCMA provides the interactive parallel coordinates plot with brushing function (especially useful for problems with more than two criteria), as well as the traditional visualization by the 2D plots generated for each criteria pair.

3. In a comprehensive MCMA, the analysis in the criteria-value space is usually integrated with analysis in the space of the model variables, especially the variables representing decisions, state of the system, diverse trajectories of the variables characterizing dynamic processes, etc. Therefore, the user can optionally specify a list of the core-model (possibly compound, i.e., indexed) variables, values of which are stored at each iteration, and provided in the CSV format in the report directory.

2.2.2. Miscellaneous features

The wall-clock time of computing the Pareto front depends on: (1) the time required for the core-model single optimization, (2) the number of criteria, and (3) the required resolution of the Pareto set representation. The resolution is improving monotonically along the iterations. Obviously, a coarse resolution is achieved much faster than a fine one. In order to help the users decide what resolution meets their needs and is worth (possibly much) longer computation time, pyMCMA periodically provides information about the characteristics of the current Pareto set representation.

The representation is poorly characterized by only a number (e.g., a maximum distance between neighbor solutions). Therefore, to provide better information about the computation progress, pyMCMA defines computation stages: at each stage the maximum (not yet processed) cuboid size is below the prescribed level. The Pareto-front representation at the end of each stage is characterized by:

- Kernel Density Estimation (KDE) plots of the distances between the neighbor solutions,
- the numbers of iterations and distinct solutions, respectively, and
- the prescribed maximum and the actual cuboid sizes.

Section 3 provides examples of the corresponding plots.

3. Illustrative example

pyMCMA functionality is illustrated by the model derived from the family of actual models developed for analysis of technology pathways of China's liquid fuel production [59–61]. The model's decision variables include trajectories of investments in the production capacities of diverse technologies that need to cover a given trajectory of demand for fuels. Technologies differ by costs (investment and operations), carbon emission (one technology captures carbon), water use, classification of the fuel produced (either fossil-based or renewable/green), as well as by many other technological attributes. The illustrative core model is composed of the actual model specification and the data defining small model instance.

Four criteria are defined as the corresponding totals of the whole planning period:

- Total cost (minimize).
- Carbon balance (difference between carbon emission and capture) (minimize).
- Water use (minimize).
- GrFuel (Green fuel)[1] (maximize).

Fig. 3 and 4 illustrate the results of the computed Pareto-front representation. The former illustrates criteria trade-offs for small/initial representation (obtained through a small number of iterations), while the latter shows the criteria trade-offs for the whole Pareto-set. The interactive brushing option is used in the latter to focus attention on a selected range of criteria achievements, which is [60, 75] in this case. The criteria performance is normalized to the common scale, where 0 and 100 correspond to the criterion's worst and best values within the Pareto-front, respectively. Due to the space considerations, we point out only selected features of the computed representation. Such plots are especially useful for analyses involving more than two criteria.

Interaction with the parallel-coordinates plot is available only during pyMCMA execution; here we can only outline its function. One can select a range of the achievements of the left-most criterion on the slider (like [60, 75] for the cost criterion in the example) to focus on the corresponding achievements of all other criteria. The achievements outside the selected range are shown in dimmed colors. In other words, the selection excluded the cheap solutions (CAF > 75) and expensive (CAF < 60) solutions. The dimmed cheap solutions have rather bad achievements of the remaining criteria. It is also interesting to note that except of very cheap solutions, the performance of the *water* and *grFuel* criteria is very good.

Another Figure generated by pyMCMA shows criteria trade-offs for all criteria pairs. Such Figure, not presented here due to the space constraints, shows the so-called knee-effect, i.e., the area in which the trade-offs rates change substantially, see e.g., [51–53].

The abovementioned Figures serve only as the initial illustration of results in the criteria space. All solutions defining the Pareto-front representation are exported in the CSV format to be available for actual, usually in-depth, problem analysis, focused on the relevance to each specific modeled problem. Moreover, the user can also request (in the analysis configuration file) the export of selected (also indexed) core-model variables. Details on the exported values are provided in the

[1] Planned to be the only fuel allowed for internal combustion engines in the EU after 2035.

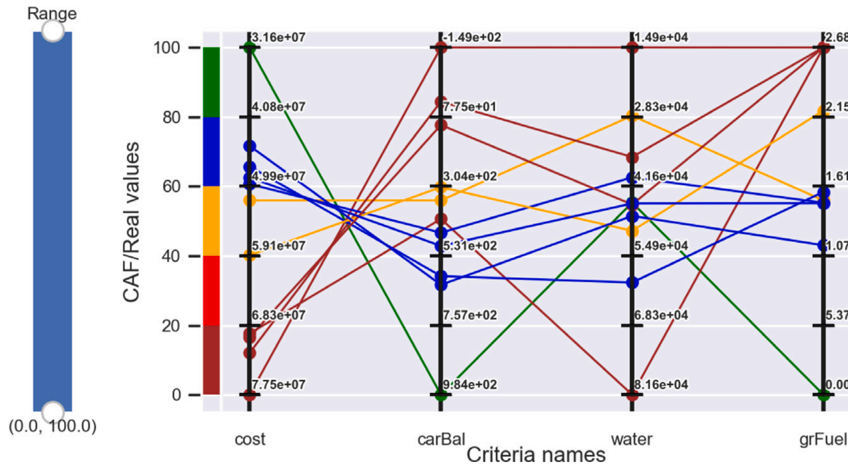

Fig. 3. Criteria trade-offs in interactive parallel-coordinates plot, initial Pareto set.

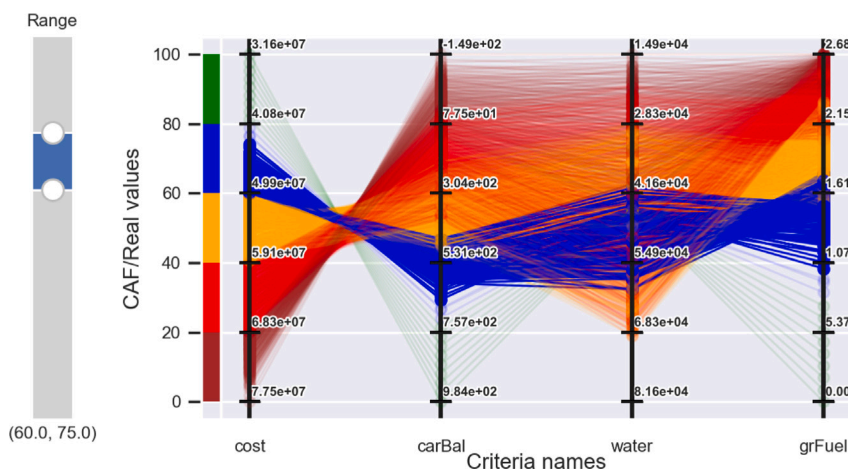

Fig. 4. Criteria trade-offs in interactive parallel-coordinates plot, brushed.

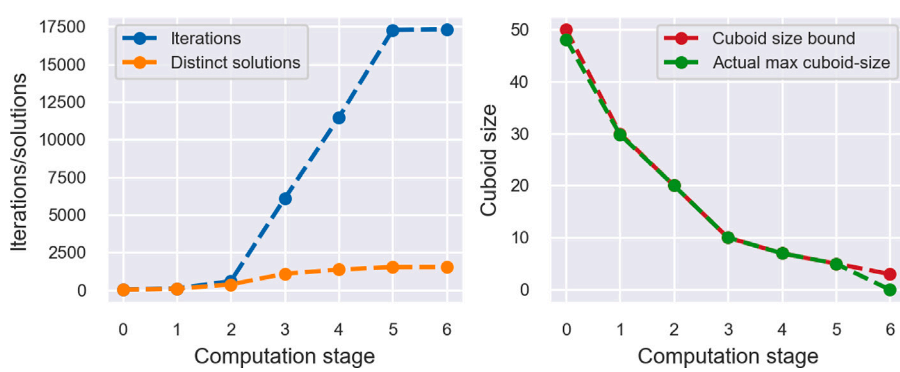

Fig. 5. Computation's progress in terms of: (Left) numbers of iterations and distinct solutions; (Right) maximum distance between neighbor solutions.

online documentation. This option greatly supports the integration of analysis in the two spaces: the criteria and the *core model*. The first provides an analysis of trade-offs between the simultaneously reachable goals for the criteria, and the second the consequences in terms of the model decision and state variables.

The computation of the Pareto-front representation for large models in high resolution (i.e., the maximum distance between neighbor Pareto solutions) may require a long time. `pyMCMA` provides information about the computation's progress in reaching predefined stages of the resolution. Fig. 5 shows in the left-side plot the number of computed distinct solutions and in the right-side plot the maximum distance between neighbor solutions.

Let us recall that each iteration provides a Pareto solution. In order to provide the uniformly distributed representation, as well as to greatly increase the computation's efficiency, `pyMCMA` filters on the fly the close solutions; therefore, the number of the found distinct solutions grows slower along the already achieved resolution of the representation. In other words, the share of duplicated solutions grows with the decreasing maximum distance between neighbor solutions.

The maximum distance between neighbor solutions alone is a rather poor measure of the representation quality. Therefore, Fig. 6 shows the

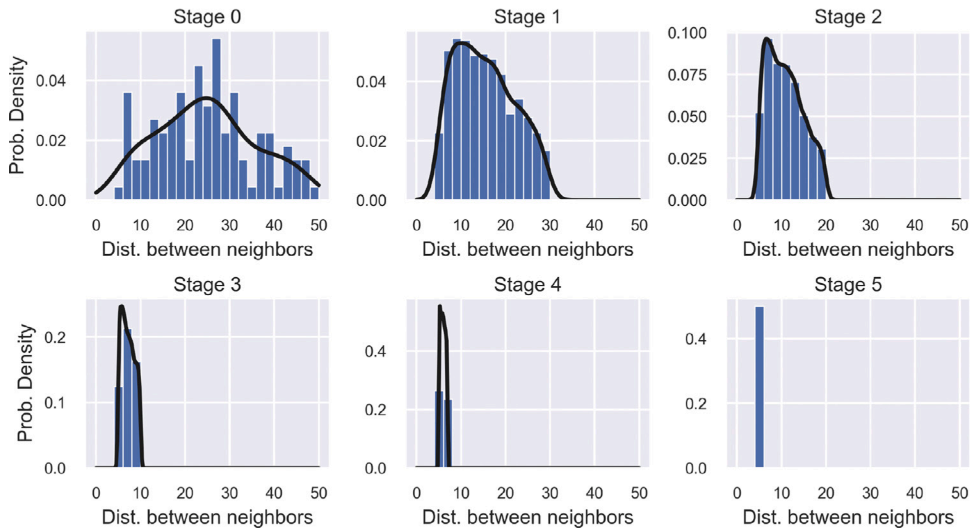

Fig. 6. KDE (Kernel Distribution Estimation) of distances between neighbor solutions along the computation's stages.

KDE (Kernel Distribution Estimation) of such distances along progressing computation stages.

4. Impact

MCMA is essential in integrated problem analysis, especially when one explores ways to simultaneously reach goals for several conflicting objectives. Such explorations consist of two interlinked types of analysis: (1) finding a suitable representation of the Pareto front, and (2) analysis of trade-offs between criteria values of diverse efficient solutions.

pyMCMA impact builds on numerous successful MCMA applications in diverse fields, documented in thousands of publications in almost all areas of science and economy. To illustrate this issue, we mention only a small sample of analyses done by coauthors of this paper in fields of: energy [59,62,63], environment and climate [6,64,65], industry [66–68], agriculture [69,70], water [71,72], negotiations [73], and medicine [74].

These, and numerous others, applications required thousands of interactive MCMA iterations to obtain just several dozens of representative Pareto solutions. This in turn required weeks of researchers' time needed for sequential specifications of preferences for each iteration. The practice of MCMA problems with more than three criteria shows that it is often very difficult to specify preferences that lead to closing gaps (defined by the achievements of one or more criteria) between neighbor Pareto solutions. Additionally, in some problems the Pareto-front consists of disconnected sets; pyMCMA recognizes such situations and reduces only the distances that can be reduced.

The main pyMCMA features characterizing its key impact potential are:

- Autonomous computation of uniformly distributed representation of the Pareto set.
- Easy, parameterization-free, and seamless integration with independently developed model instances.
- Easy storing of the data suitable for the problem-specific dedicated analysis.

5. Conclusions

The presented pyMCMA methodology and software autonomously generate Pareto-front representation for seamlessly integrated core models. The Pareto solutions have uniformly distributed criteria values. The Pareto front is effectively illustrated also for problems with more than three criteria; interactive parallel-coordinates plot with the brushing function supports analysis of criteria trade-offs in diverse regions of the Pareto front. The results in the criteria space are stored for problem-specific analysis. Optionally, the user can specify (also indexed) variables of the analyzed core model; values of such variables are stored for the dedicated, problem-specific post-MCMA analysis. pyMCMA is very easy to use through simple command-line executions and easy problem configuration in the YAML configuration file, which effectively supports diverse analyses of the provided model.

CRediT authorship contribution statement

Marek Makowski: Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Janusz Granat:** Writing – review & editing, Validation, Supervision, Software, Project administration, Methodology, Funding acquisition, Conceptualization. **Andrii Shekhovtsov:** Writing – review & editing, Visualization, Validation, Software. **Zbigniew Nahorski:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition. **Jinyang Zhao:** Writing – review & editing, Methodology, Investigation, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors gratefully acknowledge the financial support of the National Science Centre, Poland [2018/30/Q/HS4/00764], and of the National Natural Science Foundation of China (71961137012, 7214006). The pyMCMA methodology is based on the idea proposed by late Prof. Włodek Ogryczak in discussions with the authors of this paper in 2009–2010. The basic idea originated from the experiences with successful implementations of the ISAAP methodology described e.g., in [10] and applied to numerous problems, see Section 4 for examples.

References

[1] Wierzbicki A. Interactive decision analysis and interpretative computer inteligence. In: Grauer M, Wierzbicki A, editors. Interactive decision analysis. Lecture notes in economics and mathematical systems, vol. 229, Berlin, New York: Springer Verlag; 1984, p. 2–19.

[2] Sawaragi Y, Nakayama H, Tanino T. Theory of multiobjective optimization. New York: Academic Press; 1985.

[3] Steuer R. Multiple criteria optimization: Theory, computation, and application. New York: J. Wiley & Sons; 1986.

[4] Lewandowski A, Wierzbicki A. Decision support systems using reference point optimization. In: Lewandowski A, Wierzbicki A, editors. Aspiration based decision support systems: Theory, software and applications. Lecture notes in economics and mathematical systems, vol. 331, Berlin, New York: Springer Verlag; 1989, p. 3–20.

[5] Belton V, Stewart T. Multiple criteria decision analysis. An integrated approach. Boston: Kluwer Academic Publishers; 2002.

[6] Wierzbicki A, Makowski M, Wessels J, editors. Model-based decision support methodology with environmental applications. Series: mathematical modeling and applications, Dordrecht: Kluwer Academic Publishers; 2000.

[7] Figueira J, Greco S, Ehrgott M, editors. Multiple criteria decision analysis: State of the art surveys. New York: Springer; 2005.

[8] Greco S, Ehrgott M, Figueira J. Multiple criteria decision analysis: State of the art surveys. New York: Springer; 2016, http://dx.doi.org/10.1007/978-1-4939-3094-4.

[9] Ogryczak W, Lahoda S. Aspiration/reservation-based decision support — A step beyond goal programming. J Multi-Criteria Decis Anal 1992;1(2):101–17.

[10] Granat J, Makowski M. Interactive specification and analysis of aspiration-based preferences. European J Oper Res 2000;122(2):469–85.

[11] Geoffrion A. Proper effiiciency and the theory of vector maximization. J Math Anal Appl 1968;22:618–30.

[12] Yu P. Compromise solutions, domination structures, and Salukvadze's solution. J Optim Theory Appl 1974;13(3):362–78.

[13] Wierzbicki A. Basic properties of scalarizing functionals for multiobjective optimization. Math Oper forsch Stat, s. Optim 1977;8:55–60.

[14] Wierzbicki A. A mathematical basis for satisficing decision making. Math Model 1982;3(5):391–405.

[15] Stewart T. A critical survey on the status of multiple criteria decision making theory and practice. Omega 1992;20(5/6):569–86.

[16] Wierzbicki A. On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR Spektrum 1986;8:73–87.

[17] Makowski M, Wierzbicki A. Modeling knowledge: Model-based decision support and soft computations. In: Yu X, Kacprzyk J, editors. Applied decision support with soft computing. Series: studies in fuzziness and soft computing, vol. 124, Berlin, New York: Springer-Verlag; 2003, p. 3–60, draft version available from http://www.iiasa.ac.at/~marek/pubs/prepub.html.

[18] Sayin S. A procedure to find discrete representations of the efficient set with specified coverage errors. Oper Res 2003;51(3):427–36.

[19] Ruzika S, Wiecek MM. Approximation methods in multiobjective programming. J Optim Theory Appl 2005;126(3):473–501. http://dx.doi.org/10.1007/s10957-005-5494-4.

[20] Laumanns M, Thiele L, Zitzler E. An adaptive scheme to generate the Pareto front based on the epsilon-constraint method. In: Branke J, Deb K, Miettinen K, Steuer RE, editors. Practical approaches to multi-objective optimization. Dagstuhl seminar proceedings (DagSemProc), vol. 4461, Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik; 2005, p. 1–11. http://dx.doi.org/10.4230/DagSemProc.04461.6.

[21] Nakayama H, Yu Y, Yoo M. Sequential approximate multiobjective optimization using computational intelligence. Berlin, Heidelberg: Springer; 2009.

[22] Utyuzhnikov SV, Fantini P, Guenov MD. A method for generating a well-distributed pareto set in nonlinear multiobjective optimization. J Comput Appl Math 2009;223:820–41.

[23] Hartikainen M, Miettinen K, Wiecek MM. Constructing a pareto front approximation for decision making. Math Methods Oper Res 2011;73(2):209–34. http://dx.doi.org/10.1007/s00186-010-0343-0.

[24] Auger A, Bader J, Brockhoff D, Zitzler E. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. Theoret Comput Sci 2012;425:75–103.

[25] Ghosh D, Chakraborty D. A new pareto set generating method for multi-criteria optimization problems. Oper Res Lett 2014;42(8):514–21. http://dx.doi.org/10.1016/j.orl.2014.08.011.

[26] Shao L, Ehrgott M. Discrete representation of non-dominated sets in multi-objective linear programming. European J Oper Res 2016;255(3):687–98. http://dx.doi.org/10.1016/j.ejor.2016.05.001.

[27] Bazgan C, Jamain F, Vanderpooten D. Discrete representation of the non-dominated set for multi-objective optimization problems using kernels. European J Oper Res 2017;260(3):814–27. http://dx.doi.org/10.1016/j.ejor.2016.11.020, URL https://www.sciencedirect.com/science/article/pii/S0377221716309456.

[28] Paquete L, Schulze B, Stiglmayr M, Lourenço AC. Computing representations using hypervolume scalarizations. Comput Oper Res 2022;137:105349.

[29] Guerreiro AP, Cortes J, Vanderpooten D, Bazgan C, Lynce I, Manquinho V, et al. Exact and approximate determination of the pareto front using minimal correction subsets. Comput Oper Res 2023;153:106153. http://dx.doi.org/10.1016/j.cor.2023.106153.

[30] Ghosh Jauny D, Ansari QH, Ehrgott M, Upadhayay A. An infeasible interior-point technique to generate the nondominated set for multiobjective optimization problems. Comput Oper Res 2023;155:106236. http://dx.doi.org/10.1016/j.cor.2023.106236.

[31] Mesquita-Cunha M, Figueira JR, Barbosa-Póvoa AP. New ϵ-constraint methods for multi-objective integer linear programming: A pareto front representation approach. European J Oper Res 2023;306(1):286–307. http://dx.doi.org/10.1016/j.ejor.2022.07.044.

[32] Jaszkiewicz A, Słowiński R. The 'light beam search' approach – An overview of methodology applications. European J Oper Res 1999;113(2):300–14. http://dx.doi.org/10.1016/S0377-2217(98)00218-5.

[33] Deb K, Kumar A. Light beam search based multi-objective optimization using evolutionary algorithms. In: 2007 IEEE congress on evolutionary computation. 2007, p. 2125–32. http://dx.doi.org/10.1109/CEC.2007.4424735.

[34] Bhunia AK, Samanta SS. A study of interval metric and its application in multi-objective optimization with interval objectives. Comput Ind Eng 2014;74:169–78. http://dx.doi.org/10.1016/j.cie.2014.05.014.

[35] Dächert K, Klamroth K, Lacour R, Vanderpooten D. Efficient computation of the search region in multi-objective optimization. European J Oper Res 2017;260(3):841–55. http://dx.doi.org/10.1016/j.ejor.2016.05.029.

[36] Wang Y, Li L, Yang K, Emmerich MTM. A new approach to target region based multiobjective evolutionary algorithms. In: 2017 IEEE congress on evolutionary computation. 2017, p. 1757–64. http://dx.doi.org/10.1109/CEC.2017.7969514.

[37] Galuzio PP, de Vasconcelos Segundo EH, d.S. Coelho L, Mariani VC. Mobopt & multi-objective Bayesian optimization. SoftwareX 12. (2024/02/11 2020).

[38] Kaliszewski I, Miroforidis J. Cooperative multiobjective optimization with bounds on objective functions. J Global Optim 2021;79(2):369–85.

[39] Kaliszewski I, Karelkina O. Idol: A web application for mixed integer linear multiobjective optimization. SoftwareX 19. (2024/02/11 2022).

[40] Kaliszewski I, Miroforidis J. Probing the pareto front of a large-scale multiobjective problem with a mip solver. Oper Res 2022;22(5):5617–73. http://dx.doi.org/10.1007/s12351-022-00708-y.

[41] Sayın S. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Math Program 2000;87(3):543–60. http://dx.doi.org/10.1007/s101070050011.

[42] Faulkenberg SL, Wiecek MM. On the quality of discrete representations in multiple objective programming. Optim Eng 2010;11(3):423–40.

[43] Morse J. Reducing the size of the nondominated set: Pruning by clustering. Comput Oper Res 1980;7(1):55–66. http://dx.doi.org/10.1016/0305-0548(80)90014-3.

[44] Taboada HA, Baheranwala F, Coit DW, Wattanapongsakorn N. Practical solutions for multi-objective optimization: An application to system reliability design problems. Reliab Eng Syst Saf 2007;92(3):314–22. http://dx.doi.org/10.1016/j.ress.2006.04.014.

[45] Brusco MJ. Partitioning methods for pruning the pareto set with application to multiobjective allocation of a cross-trained workforce. Comput Ind Eng 2017;111:29–38. http://dx.doi.org/10.1016/j.cie.2017.06.035.

[46] Petchrompo S, Coit DW, Brintrup A, Wannakrairot A, Parlikad AK. A review of pareto pruning methods for multi-objective optimization. Comput Ind Eng 2022;167. http://dx.doi.org/10.1016/j.cie.2022.108022.

[47] Taboada HA, Coit DW. Data clustering of solutions for multiple objective system reliability optimization problems. Qual Technol Quant Manag 2007;4(2):191–210. http://dx.doi.org/10.1080/16843703.2007.11673145.

[48] Messac A, Mattson CA. Generating well-distributed sets of pareto points for engineering design using physical programming. Optim Eng 2002;3(4):431–50.

[49] Gong W, Duan Q, Li J, Wang C, Di Z, Ye A, et al. Multiobjective adaptive surrogate modeling-based optimization for parameter estimation of large, complex geophysical models. Water Resourc Res 2016;52(3):1984–2008. http://dx.doi.org/10.1002/2015WR018230, arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2015WR018230.

[50] Lou Y, Wang S. Approximate representation of the pareto frontier in multiparty negotiations: Decentralized methods and privacy preservation. European J Oper Res 2016;254(3):968–76. http://dx.doi.org/10.1016/j.ejor.2016.04.060.

[51] Branke J, Deb K, Dierolf H, Osswald M. Finding knees in multi-objective optimization. In: Yao X, Burke EK, Lozano JA, Smith J, Merelo-Guervós JJ, Bullinaria JA, Rowe JE, Tiňo P, Kabán A, Schwefel H-P, editors. Parallel problem solving from nature - PPSN vIII. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004, p. 722–31.

[52] Zhang X, Tian Y, Jin Y. A knee point-driven evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 2015;19(6):761–76. http://dx.doi.org/10.1109/TEVC.2014.2378512.

[53] Sudeng S, Wattanapongsakorn N. A knee-based multi-objective evolutionary algorithm: An extension to network system optimization design problem. Cluster Comput 2016;19(1):411–25.

[54] Nagar D, Ramu P, Deb K. Interpretable self-organizing maps (isom) for visualization of pareto front in multiple objective optimization. In: Ishibuchi H, Zhang Q, Cheng R, Li K, Li H, Wang H, Zhou A, editors. Evolutionary Multi-Criterion Optimization. Cham: Springer International Publishing; 2021, p. 645–55.

[55] Yadav D, Nagar D, Ramu P, Deb K. Visualization-aided multi-criteria decision-making using interpretable self-organizing maps. European J Oper Res 2023;309(3):1183–200. http://dx.doi.org/10.1016/j.ejor.2023.01.062.

[56] Bynum ML, Hackebeil GA, Hart WE, Laird CD, Nicholson BL, Siirola JD, et al. Pyomo–optimization modeling in Python, Vol. 67. third ed.. Springer Science & Business Media; 2021.

[57] Makowski M. A structured modeling technology. European J Oper Res 2005;166(3):615–48.

[58] Rawls J. A theory of justice. London, Oxford, New York: Oxford University Press; 1973.

[59] Zhao J, Yu Y, Ren H, Makowski M, Granat J, Nahorski Z, et al. How the power-to-liquid technology can contribute to reaching carbon neutrality of the China's transportation sector? Energy 2022;261:125058. http://dx.doi.org/10.1016/j.energy.2022.125058, published online: August 22 2022.

[60] Ding B, Makowski M, Nahorski Z, Ren H, Ma T. Optimizing the technology pathway of China's liquid fuel production considering uncertain oil prices: A robust programming model. Energy Econ 2022;115:106371. http://dx.doi.org/10.1016/j.eneco.2022.106371, published online: October 21 2022.

[61] Ding B, Makowski M, Zhao J, Ren H, Zakeri B, Ma T. Analysis of technology pathway of China's liquid fuel production with consideration of energy supply security and carbon price. J Manag Sci Eng 2023;8:1–14. http://dx.doi.org/10.1016/j.jmse.2022.07.002, published online: August 7 2022.

[62] McCollum D, Krey V, Riahi K, Kolp P, Makowski M, Schreck B. The IIASA energy-multicriteria analysis ENE-mCA policy tool, user manual. Laxenburg, Austria: International Institute for Applied Systems Analysis; 2012, http://www.iiasa.ac.at/web-apps/ene/GeaMCA.

[63] Lehtveer M, Makowski M, McCollum D, Strubegger M. Multi-criteria analysis of nuclear power in the global energy system: Assessing trade-offs between simultaneously attainable economic, environmental and social goals. Energy Strategy Rev 2015;8:45–55. http://dx.doi.org/10.1016/j.esr.2015.09.004.

[64] McCollum D, Krey V, Riahi K, Kolp P, Grübler A, Makowski M, et al. Climate policies can help resolve energy security and air pollution challenges. Clim Change 2013;2(109):479–94. http://dx.doi.org/10.1007/s10584-013-0710-y.

[65] Parkinson S, Makowski M, Krey V, Sedraoui K, Almasoud A, Djilali N, et al. A multi-criteria model analysis framework for assessing integrated water-energy system transformation pathways. Appl Energy 2018;210(15 January):477–86. http://dx.doi.org/10.1016/j.apenergy.2016.12.142.

[66] Ren H, Zhou W, Makowski M, Yan H, Yu Y, Ma T. Incorporation of life cycle emissions and carbon price uncertainty into the supply chain network management of PVC production. Ann Oper Res 2021;300:601–20. http://dx.doi.org/10.1007/s10479-019-03365-1, published online: September 11 2019.

[67] Ren H, Zhou W, Makowski M, Zhang S, Yu Y, Ma T. A multi-criteria decision support model for adopting energy efficiency technologies in the iron and steel industry. Ann Oper Res 2022. http://dx.doi.org/10.1007/s10479-022-04548-z, Published online: February 7 2022.

[68] Ogryczak W, Sliwinski T. On equitable approaches to resource allocation problems: The conditional minimax solutions. J Telecommun Inf Technol 2002;2002(3):40–8.

[69] Antoine J, Fischer G, Makowski M. Multiple criteria land use analysis. Appl Math Comput 1997;83(2–3):195–215.

[70] Fischer G, Granat J, Makowski M. AEZWIN an interactive multiple-criteria analysis tool for land resources appraisal. World soil resources reports, vol. 87, Laxenburg, Austria, and Rome, Italy: International Institute for Applied Systems Analysis, and Food and Agriculture Organization of the United Nations; 1999.

[71] Makowski M, Somlyódy L, Watkins D. Multiple criteria analysis for water quality management in the Nitra basin. Water Resourc Bull 1996;32(5):937–51.

[72] Makowski M, Somlyódy L. River basin water quality management. Math Model Theory Appl 2000;9:311–32.

[73] Wierzbicki AP, Kruś L, Makowski M. The role of multi-objective optimization in negotiation and mediation support. Theory and Decision 1993;34:201–14.

[74] van Haveren R, Breedveld S, Keijzer M, Voet P, Heijmen B, Ogryczak W. Lexicographic extension of the reference point method applied in radiation therapy treatment planning. European J Oper Res 2017;263(1):247–57. http://dx.doi.org/10.1016/j.ejor.2017.04.062.