Structured Pareto Front Representation

Marek Makowski [a,b,*], Janusz Granat [c,d], Zbigniew Nahorski [b,g], Andrii Shekhovtsov [d,e], Hongtao Ren [f], Jinyang Zhao [f,h]

[a] *International Institute for Applied Systems Analysis, Schlossplatz 1, 2361, Laxenburg, Austria*
[b] *Systems Research Institute, Polish Academy of Sciences, Newelska 6, 00-447, Warsaw, Poland*
[c] *Warsaw University of Technology, Nowowiejska 15/19, 00-665, Warsaw, Poland*
[d] *National Institute of Telecommunications, Szachowa 1, 04-894, Warsaw, Poland*
[e] *West Pomeranian University of Technology in Szczecin, Żołnierska 49, 71-210, Szczecin, Poland*
[f] *School of Business, East China University of Science and Technology, Meilong Road 130, 200237, Shanghai, China*
[g] *WIT Academy, Newelska 6, 00-447, Warsaw, Poland*
[h] *PetroChina Planning and Engineering Institute, China National Petroleum Cooperation, West Zhixin Road 3, 100083, Beijing, China*

Abstract

The paper presents a novel methodology for autonomous generation of the Pareto-Front Representation (PFR) of Linear Programming (LP) models. Following the Structured Modeling principles, the developed approach supports multiple-criteria analysis of independently developed diverse LP models. The analysis is done by seamless linkage with the dedicated implementation of reusable Multiple Objective Programming (MOP) model, which represents the developed method of PFR's generation. The MOP is a small LP model, therefore the linked models are optimized by an LP solver.

The methodology enables autonomous (i.e., parametrization-free) generation of a sequence of single-objective LP optimization tasks, each providing a Pareto-efficient solution that improves the PFR distribution in terms of the distances between neighbor Pareto solutions. Furthermore, the method's recent enhancement by the structured PFR generation has two key advantages: it dramatically decreases the computation time, and it substantially improves the distribution of the PFR's elements. Moreover, the method properly and efficiently computes the extreme points of the PF also when optimization of a criterion has non-unique solution.

The approach supports objective Multiple Criteria Model Analysis (MCMA), i.e., equitable treatment of all criteria in the whole space of Pareto solutions. The paper provides examples of research projects in various fields of science, which required effective support for generation of preference-free PFR. Such analysis is also helpful for preference-guided MCMA because the provided PFR clusters are a good starting point for exploration of diverse Regions of Interest (ROI) based on diverse simultaneously reachable goals for conflicting criteria.

Keywords:

[*] Corresponding author, marek@iiasa.ac.at (Marek Makowski)

Multiple objective programming, Multiple-criteria model analysis, Pareto-front representation, Structured Modeling, Clustering.

1. Introduction

Scientific analysis of any nontrivial problem requires integrations of the relevant knowledge into a corresponding mathematical programming model, further on called *the core (aka substantive) model*. Integrated analysis of such models creates knowledge, in particular on diverse combinations of simultaneously attainable goals for several objectives defined in the corresponding model. The objectives are typically pairwise either conflicting or synergetic; these relations might hold in only parts of the objective space. Moreover, objectives are often defined in different measurement units and take values that differ by many orders of magnitude.

Multiple-Criteria Model Analysis (MCMA) provides methods and tools effectively supporting such model analysis. The MCMA is based on the concept of the Pareto (aka efficient or non-dominated) solution, i.e., such a solution of the underlying model that no other solution exists with at least equally good values of all criteria. Any Pareto solution is preferred over a subset of non-Pareto (dominated by that solution) feasible solutions of the corresponding core model. Pareto solutions are objectively incomparable; the choice of a particular solution depends on (idiosyncratic, subjective) preferences for trade-offs between simultaneously attainable goals for often conflicting objectives. The trade-off in this context means how much one compromises the achievement of at least one objective in order to improve the achievement of another objective.

The number of efficient solutions of continuous models is infinite; such a number for discrete models is finite but typically (very) large. Therefore, the prevailing MCMA paradigms focus on the analysis of a small subset of Pareto solutions that best fit the idiosyncratic preferences defined by diverse users. Such approaches are well researched and supported by diverse methods of preferences specification, and by the corresponding software, typically enabling interactive analysis.

Much less attention has been paid to the issue dealt with by this paper. Namely, an objective (i.e., preference-free) model analysis aimed at exploration of the whole set of Pareto solutions, often called the Pareto-Front (PF). Such analysis supports science-based problem analysis, which requires objectivity, in particular: (1) equitable treatment of objectives representing different measures of diverse problem's solutions, and (2) analysis of trade-offs between simultaneously attainable goals for the whole ranges of values of all objectives.

Computation of all PF solutions is often impractical. Therefore, one typically focuses on the Pareto Front Representation (PFR) composed of a finite number of Pareto solutions that cover the whole PF and are evenly distributed.[1]

[1]Abbreviations frequently used in this paper: ASF - Achievement Scalarizing Function; A/R - Aspiration/Reservation (either the method or a pair of the RFPs); CAF - Criterion Achievement Function; LP - Linear Programming; MCMA - Multiple Criteria Model Analysis; PF - Pareto Front; PFR - Pareto

We point out that the preference-free PFR not only supports scientific problem analysis, but also provides a good basis for decision-making support. Namely, one can explore the solutions available in the PF regions corresponding to idiosyncratic preferences. In particular, one can examine in the interactive MCMA the neighbors of solutions that match diverse individual preferences.

1.1. Motivation

The development of the presented methodology was motivated by extensive experience with applying MCMA in numerous research activities dealing with diverse actual complex problems. The authors of this paper have performed such analyses using modular implementations of the preferences specification described in Granat and Makowski (2000). We provide only a sample of applications in diverse fields: agriculture Antoine et al. (1997); Fischer et al. (1999), water Makowski et al. (1996); Makowski and Somlyódy (2000), energy, water and climate nexus Parkinson et al. (2018), energy Lehtveer et al. (2015); Zhao et al. (2022), and industry Ren et al. (2021, 2023). Obtaining a PFR suitable for objective problem analysis required, for each of these cases, many hundreds of interactive specifications of preferences, which always took several weeks. Moreover, most of the received solutions were too close to each other to actually contribute to the desired analysis of trade-offs between partly conflicting and partly synergetic criteria that cover the full range of criterion values. Next, the sequential specification of the preferences aimed at closing gaps between the criteria achievements of neighbor solutions became more and more time-consuming, as well as it required specific analytical skills often not possessed by the researchers developing the corresponding models. Additionally, lessons from these analyses triggered the desire to provide diverse functionalities to improve the research efficiency; namely, to support integrated model analysis that combines the results of MCMA, typically done in the criteria space, with problem-specific posterior (post-optimization) analysis needed in the space of the corresponding core-model variables.

We point out the qualitative differences between presentations of new methods, typically in the OR-focused publications, and their actual applicability. Most publications on the MCMA methodology use for the method illustration rather tiny models, and many consider only two criteria. Moreover, most publications amalgamate the considered MCMA method with the illustrating models. However, models of actual problems are usually large and typically involve analysis of more than two criteria. Furthermore, the model's developers often are not MCMA specialists. Therefore, MCMA tools need to be available in ways similar to optimization solvers, i.e., be easy to use with optional parameter setting, without the need of exploring the underlying methodology. Finally, as MCMA typically requires hundreds of optimization runs, the PFR generation needs to be efficient, as well as flexible, i.e., also effectively support the analysis of models whose optimization requires a long execution time.

Front Representation; RFP - Reference Point.

1.2. Scope of the paper

The above summarized demand of efficient preference-free MCMA triggered the development of the open source software package called `pyMCMA`, which provides an effective autonomous generation of evenly distributed PFR of independently developed core models. The implementation and availability of `pyMCMA` is documented in Makowski et al. (2024, 2025).

This paper presents the methodology underlying the `pyMCMA`, which could only be sketched in the short, software description focused, papers. Additionally, the paper describes the newly developed method for the structured PFR generation, which substantially improves the PFR's characteristics and dramatically decreases the computation time.

The method has been applied to diverse Linear Programming (LP) models developed within research activities. Such models are typically designed for single-criterion optimization-based analyses. Therefore, we explain the mathematical programming model architecture seamlessly integrating such core models with the module implementing the presented MCMA methodology.

1.3. Preliminaries

1.3.1. Notations

General notations follow the common practice. In particular:

- Lower- and upper-case single letter denote an index and the corresponding set, respectively; in particular, $i \in I$ are criteria indices.
- The variables written in the ***bold slanted*** font represent a vector (or a matrix) of the corresponding entities. For example: variables $\boldsymbol{x}$, functions $\boldsymbol{f}(\cdot)$, matrix of parameters $\boldsymbol{A}$.
- For comparison of two different solutions represented by vectors composed of the corresponding criteria values $\boldsymbol{v}$ and $\boldsymbol{z}$, we apply the following dominance and indifference relations:
  - ⋆ $\boldsymbol{z} \succ \boldsymbol{v}$ (equivalently $\boldsymbol{v} \prec \boldsymbol{z}$) means that $\boldsymbol{z}$ dominates (is preferred over) $\boldsymbol{v}$, i.e.,:

$$z_i \leq v_i \;\; \forall i \in I^- \quad \text{and} \quad z_i \geq v_i \;\; \forall i \in I^+, \tag{1}$$

    where I^- and I^+ are sets of indices of criteria minimized and maximized, respectively.
  - ⋆ $\boldsymbol{v} \sim \boldsymbol{z}$ means that $\boldsymbol{z}$ and $\boldsymbol{v}$ are indifferent, i.e., one cannot objectively judge which one is better.
- $\boldsymbol{p} \in P$: a Pareto solution $\boldsymbol{p}$ and the set of such solutions, respectively. Depending on the context, P can denote either the PF (the set of all efficient solutions, which for LP models is unknown) or its current representation denoted by PFR.
- $\boldsymbol{u}$: Utopia point, a vector composed of the criteria's best values.
- $\boldsymbol{n}$: Nadir point, a vector composed of the criteria's worst (within the PF) values.

The notations of entities used locally are explained in the corresponding places.

1.3.2. Problem formulation

We consider the specific case of the Multiple Objective Optimization (MOO) problem; namely, the Multiple Objective Linear Program (MOLP) as a generalization of the standard LP program with n variables and m constraints by its extension through defining

k objectives (aka criteria, outcomes):

$$(\hat{\boldsymbol{q}},\ \hat{\boldsymbol{x}}) = \arg\ \underset{\boldsymbol{q}\in Q}{\text{opt}}\ \{\boldsymbol{q} \in Q = \{\boldsymbol{q} = \boldsymbol{B}\cdot\boldsymbol{x}\} \quad \text{and} \quad \boldsymbol{x} \in X = \{\boldsymbol{x} : \underline{\boldsymbol{b}} \leq \boldsymbol{A}\cdot\boldsymbol{x} \leq \overline{\boldsymbol{b}}\}\ \} \tag{2}$$

where:

- $(\hat{\boldsymbol{q}},\ \hat{\boldsymbol{x}})$ - optimal solutions in the criteria and the model's other variables spaces, respectively;
- `opt` - an optimization criterion, e.g., `max` or `min` operator, an achievement scalarizing function;
- $\boldsymbol{q} \in \mathbb{R}^k$ - variables representing criteria;
- $\boldsymbol{x} \in \mathbb{R}^n$ - variables (decision, control, state, auxiliary) of the LP substantive model;
- $\boldsymbol{B} \in \mathbb{R}^{k\times n}$ - matrix of coefficients defining criteria (in single-criterion LP model analysis the $\boldsymbol{B}$ matrix consists of only one row, traditionally called the cost coefficients);
- X - set of feasible solutions of the LP problem;
- Q - set of feasible criteria values;
- $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ - matrix of the LP model coefficients;
- $\underline{\boldsymbol{b}} \in \mathbb{R}^m$ and $\overline{\boldsymbol{b}} \in \mathbb{R}^m$ - vectors of, respectively, lower and upper bounds of the LP model relations.

The set P of Pareto solutions $\boldsymbol{p}$ (aka the PF) can be defined by:

$$P = \{p \in Q : \{z \in Q : \boldsymbol{p} \prec \boldsymbol{z}, \boldsymbol{z} \neq \boldsymbol{p}\} = \emptyset\}, \tag{3}$$

where the dominance relation $\prec$ is defined by (1). If $\hat{\boldsymbol{p1}} \neq \hat{\boldsymbol{p2}}$ are two Pareto optimal solutions, then $\hat{\boldsymbol{p1}} \sim \hat{\boldsymbol{p2}}$. In plain language, one says that a feasible solution of (2) is Pareto optimal, if there is no other feasible solution $\boldsymbol{p3} \in Q$ with a better value of at least one criterion and equally good values of all other criteria.

1.4. State of the art

Diverse paradigms related to the MCMA have been extensively discussed since the 1950s, in countless books and articles. A survey of this huge legacy is beyond the scope of this paper; it is, however, provided by Ehrgott et al. (2026). Here, we focus on the methods relevant to the above-defined scope of this paper. Therefore, we refrain from discussing methods: (1) specific for analysis of other than LP classes of models, in particular, mixed-integer, integer, combinatorial, nonlinear, (2) requiring parametrization by the users or providing characteristics of the analyzed model, and (3) applicable only to bi-criteria analysis.

The MCMA proliferation has been greatly enhanced by the parallel developments of modeling paradigms and the corresponding modeling environments, in particular the Structured Modeling methodology Geoffrion (1987, 1989a,b); Geoffrion and Krishnan (2001) and the Structured Modeling Technology Makowski (2005), as well as by model-based decision support methods and applications, see e.g., Wierzbicki et al. (2000); Makowski and Wierzbicki (2003).

Most MCMA methods are based on he original Pareto (1896) concept of efficient solutions combined with several related mathematical concepts, in particular, the concepts dealing with dominance, efficiency, and cones, see e.g., Geoffrion (1968); Yu (1974); Wierzbicki (1977, 1982, 1986). Comprehensive presentations of such methods and the corresponding software tools developed in the XX century, partly by researchers working in parallel, including Grauer and Wierzbicki (1984); Sawaragi et al. (1985); Yu (1985); Steuer (1986); Lewandowski and Wierzbicki (1989); Climaco (1997); Stewart and van den Honert (1998); Wierzbicki et al. (2000). These were followed by numerous publications mainly extending diverse methods supporting multiple-criteria analysis guided by the user preferences. As examples of the latter, we only mention Belton and Stewart (2002); Figueira et al. (2005); Greco et al. (2016), as well as the recent comprehensive review by Ehrgott et al. (2026), and refer to the rich collections of references therein.

Further on, we review only selected publications that are closely related to the topics of the PFR generation. Ruzika and Wiecek (2005) provided the first comprehensive survey of over 50 methods for Pareto-Front representations or approximations. Numerous publications on related topics followed in the last two decades. We focus rather on PFR (by a finite discrete set of sequentially generated Pareto solutions) than on PF approximations because the former clearly better fits the needs summarized in Section 1.1. Similar judgment, also in the context of supporting decision making, is shared by e.g., Faulkenberg and Wiecek (2010). We start with reviewing the quality measures of discrete PFR in Section 1.4.1. Next, we focus on the methods that (1) provide one Pareto solution for each optimization run, and (2) organize such optimizations into a sequence leading to evenly distributed Pareto solutions covering the whole PF. We discuss these two topics in Sections 1.4.2 and 1.4.3, respectively.

1.4.1. Quality measures of discrete PFR

Most of the known quality measures address in various ways the original postulate of Benson and Sayin (1997) paraphrased here as: *"a good PFR needs to contain a reasonable number of points, should not miss large portions of PF, and should not contain points that are very close to each other."* Faulkenberg and Wiecek (2010) provided an extensive overview of diverse quality measures characterizing a "good" discrete PFR. Wang et al. (2017) reviewed and compared numerous alternative or complementary measures. Diverse measures are discussed in detail in many other publications, e.g., in Sayın (2000); Deb et al. (2002); Messac and Mattson (2002); Shao and Ehrgott (2016).

The most commonly used measures include: cardinality, coverage, uniformity (aka spacing, evenness, distribution), and crowding distance. Some of the measures, originally developed for either evaluating evolutionary algorithms or benchmarking particular methods of the PFR generation, are adapted for characterizing autonomously generated PFR. Sec. 2 provides details of such adaptation.

1.4.2. Achievement Scalarizing Function

A Pareto solution can be provided by single-criterion optimization of the core model in which the original objective function (defined for single criterion optimization) is replaced by a function representing specific preferences for trade-offs between multiple objectives, as well as guarantees that the resulting solution is Pareto efficient. Such a function is often called *Aggregated Objective Function* or *Achievement Scalarizing Function (ASF)*. For the reasons explained in Section 2, we use the latter name.

In order to simplify the discussion, we assume the following local notation: (1) all criteria are maximized; (2) vector $\boldsymbol{q} = \boldsymbol{f}(\cdot)$, where $\boldsymbol{f}(\cdot)$ are linear functions defining the corresponding criteria; (3) vectors $\boldsymbol{n}$ and $\boldsymbol{u}$ represent the Nadir and Utopia, respectively; and (4) vectors $\bar{\boldsymbol{q}}$ and $\boldsymbol{w}$ stand for the specified RFP and weights, respectively.

We denote the ASF by $asf(\boldsymbol{q}, \phi)$, where $\boldsymbol{q}$ stands for variables defining criteria, in our case defined by (2), and parameters ϕ represent the preferences. Whenever the context is obvious, we use the $asf(\cdot)$ shorthand.

Many diverse ASF's specifications are described in most of the above cited publications, as well as in the countless numbers of other publications. We start the ASF's review with the probably oldest, and still popular, the linear scalarization method, aka Weighted Sum Method (WSM):

$$asf(\boldsymbol{q}, \boldsymbol{w}) = \sum_{i \in I} w_i \cdot q_i, \quad w_i \geq 0, \; \sum_{i \in I} w_i = 1. \tag{4}$$

However, the WSM has the following key undesired properties:

- Weights w_i aggregate the necessary (PF ranges often of range of several orders of magnitude) criteria scaling with preferences, which often causes problems with the PF exploration.
- WSM provides poor control of the PF exploration of LP models because: (1) it gets solutions on only the vertices of the corresponding simplex; therefore, solutions on the edges and facets of the simplex are never found; (2) often even for large changes of weights, the solution remains at a vertex while a small change of weights causes jumps to another vertex.
- WSM has a hidden impracticable requirement of criteria independence; violation of this requirement results in the so-called double-counting of effects of synergetic criteria.
- It is expected that an increase of a weight results in an improvement of the achievement of the corresponding criterion. However, this is not always true, see Nakayama (1994); Nakayama et al. (2009) for a simple example showing that the WSM is actually counter-intuitive.

Many successfully applied ASFs are parametrized by the so-called Reference Point (RFP) consisting of the desired criteria values. A comprehensive discussion of the RFP theoretical background and of the mathematical properties of the corresponding ASFs are presented in Wierzbicki (1977, 1980, 1982, 1986), as well as in many other publications. The detailed discussion of this methodology and its relevance to supporting modern decision making, tools for its implementation, and diverse applications are presented e.g., in Wierzbicki et al. (2000).

Generally, a RFP represents preferences, i.e., the desired combination of the criteria values. A RFP is either not attainable (i.e., dominates all Pareto solutions) or it is dominated by at least one Pareto solution or it is equal to a Pareto solution. A single RFP alone extremely rarely determines a Pareto solution. Therefore, the ASF specification typically includes either another provided parameter (in particular, a second RFP or weights or a direction) or an

embedded rule for computing its substitution.
The most popular ASF is defined by the augmented weighted Chebyshev (aka Tchebyscheff) scalarizing function, maximization of which provides an ϵ-properly Pareto solution:

$$asf(\boldsymbol{q}, \bar{\boldsymbol{q}}) = \min_{i \in I}(w_i \cdot (q_i - \bar{q}_i)) + \epsilon \cdot \sum_{i \in I} w_i \cdot (q_i - \bar{q}_i), \quad w_i = 1/|u_i - n_i|, \tag{5}$$

where $\bar{q}_i, u_i, n_i$ are the components of the RFP, Utopia, and Nadir, respectively, and ϵ is a small positive number. The relation (5) is a general definition equivalent (or similar) to many widely published, diverse definitions. In some of them, either $\boldsymbol{u}$ or $\boldsymbol{n}$ parameter is replaced by the $\bar{\boldsymbol{q}}$. We point out the main problems in using the ASF defined by (5), especially for the analysis of models of actual problems:

1. Computation of the correct Nadir for more than two criteria is not only time-consuming but often requires a dedicated procedure see e.g., Isermann and Steuer (1987), Benson and Sayin (1997), Alves and Costa (2009).
2. The popular approach of determining the Nadir from the so-called Payoff table (resulting from individual optimization of each criterion) often provide incorrect Nadir values, see e.g. example from Deb et al. (2009).
3. For many practical problems, some criteria values differ (within the PF) by many orders of magnitudes, see e.g., Lehtveer et al. (2015). In such cases, weights w_i used in the ASF (5) often cause numerical problems.

Note, that $\boldsymbol{w}$ defined within (5) serve as criteria scaling. In some approaches, the weights represent also the preferences; in such cases, they are defined accordingly.
Diverse specific ASF definitions using the RFP are reviewed in e.g., Ruiz et al. (2008); Ehrgott et al. (2026). The problems with applications of the above-summarized ASF parametrizations to real-case models motivated the development of the method parameterizing the ASF by a pair of provided RFPs, namely the A/R method, where A and R stand for the preferred criteria values that one wants to achieve and avoid, respectively. In other words, preferences are expressed as the corresponding ranges of values for each criterion. The A/R method, developed in 1980s for an interactive multiple-criteria model analysis, provides complete control in navigating the whole PF of also nonconvex and/or disjoint PF. A comprehensive discussion of the theoretical background of RFP approach in general, and of the A/R method in particular can be found e.g., in Granat and Makowski (2000); Wierzbicki et al. (2000); Makowski and Wierzbicki (2003). The relations of the A/R method to the commonly known Goal Programming are summarized by Ogryczak and Lahoda (1992). Such ASF has been successfully applied in MCMA of diverse complex models, as well as in the autonomous and parametrization-free PFR generation presented in detail in Sec. 2. Alternative approaches to ASF parametrization by pairs of RFPs are summarized in Sec. 1.4.3.
The ASF can also be parametrized by a single RFP and a direction. The direction can either be assumed to be diagonal (aka equidistant), see e.g., Steuer (1986) or implied by the ($\boldsymbol{u}$, $\boldsymbol{n}$) points. Moreover, diverse methods for defining directions and the corresponding ASFs can be found e.g., in Pascoletti and Serafini (1984); Benson and Sayin (1997); Eichfelder (2009).

1.4.3. Generation of PFR

We consider PFR consisting of a set of Pareto solutions provided by optimizations of sequentially parametrized ASF. In other words, a PFR of a given model is determined by the selection of: (1) a particular ASF, and (2) the corresponding procedure of its sequential parameterizations. The corresponding implementation should result in the desired characterization of the resulting PFR, as well as be computationally efficient.

Measures of the quality of a discrete representation of the Pareto set have been discussed in e.g., Sayın (2000); Faulkenberg and Wiecek (2010). Messac et al. (2003) summarized the desired properties of PFR generation methods: (1) the method should generate an even set of Pareto points in the design space and not neglect any region, (2) the method should have the ability to generate all available Pareto solutions, (3) the method should generate only Pareto solutions, and finally (4) the method should be relatively easy to apply.

The efficient sequencing of the ASF parametrization, focused on the autonomous PFR generation and having the above-specified desired properties, is presented in detail in Section 2. Here we summarize alternative approaches.

Messac et al. (2003) overviews and compares against the above properties five popular methods: (1) the physical programming (PP), (2) the normal boundary intersection (NBI), (3) the normal constraint (NC), (4) the weighted sum (WS) and, (5) the compromise programming (CP). The author claims that only the PP method has all the desired properties. Utyuzhnikov et al. (2009) describe another PP method. However, the PP methods require parameterizations depending on the analyzed model. Therefore, they are not suitable for autonomous generation of the PFR.

We also mention other methods of RFP generation, although none of them is suitable for autonomous PFR generation. Namely, the Pascoletti-Serafini (PS) method discussed by Pascoletti and Serafini (1984); Eichfelder (2009), the revised normal boundary intersection (RNBI) method by Shao and Ehrgott (2016), as well as methods combining the RFP generation with DEA, see Yun et al. (2004).

Other approaches to the PFR generation are based on the ϵ-constraint method. Eichfelder (2009) proves that use of this method may result in the generation of optimization problems that are either infeasible or provide weekly Pareto solutions; therefore, the method needs adaptive parameter control. A recent advancement of this method by Mesquita-Cunha et al. (2023) shows the complexity of such a control, which makes it impractical for autonomous PFR generation.

Finally, we comment briefly on the population-based approaches (as well as on earlier methods known as evolutionary or genetic algorithms) that are widely used for PF approximation, primarily of non-linear models. Ehrgott et al. (2026) provides a comprehensive review of also these approaches, including pointing out their notable shortcoming: *"Namely, they approximate the PF rather than systematically proving optimality. This is mainly due to randomized evolutionary operators, finite computational resources, and lack of definitive stopping criteria that ensure the entire PF has been found."* Earlier, Nakayama et al. (2009) pointed out the two main issues of these methods, namely: (1) how to guide individuals to the actual PF, and (2) how to keep the diversity of the PFR. Complementary explanations of the population-based approaches to the generation of the PFR, in particular of: the required computational resources (especially for large models), the sensitivity to the parameter settings, and convergence to a well-distributed PFR (especially for complex or high-dimensional problems) are discussed in e.g., Coello et al. (2007); Nakayama et al. (2009); Chen et al. (2011); Márquez-Vega et al. (2024); Saini et al. (2025). We conclude that for MOLP population-based approaches are clearly less suitable than those based on the sequential generation of Pareto solutions by optimization of a selected ASF.

1.5. Contribution

The presented novel methodology enables efficient, parametrization-free, autonomous generation of evenly distributed PFR of the independently developed LP model. The modular structure enables the PFR generation in a similar way as the usage of an optimization solver. Furthermore, the method's recent enhancement by the structured PFR generation has two key advantages compared to its previous version: (1) it dramatically decreases the computation time (from hours to seconds), and (2) it substantially improves the distribution of the PFR's elements.

The method effectively integrates the two features that have not been used in other methods of PFR generation: (1) structured selection of pairs of neighbor Pareto solutions, and (2) iterative computation of an ϵ-properly Pareto solution that sequentially decreases the largest distance between the neighbor solutions of the already generated PFR.

Moreover, the method properly and efficiently computes the extreme points of the PF also when optimization of a criterion has non-unique solution, i.e., there are many different solutions with the same (best) criterion value.

Section 5 provides details on the features of the presented method.

1.6. Content of the paper

The remaining part of the paper is structured as follows. Section 2 presents the methodology, the main part of the paper. It is followed by the discussion of case studies in Section 3. The posterior analysis and the summary of the approach are discussed in Sections 4 and 5, respectively. Section 6 concludes. Additionally, Appendix A provides a real-case example of non-unique solutions of a criterion (selfish) optimization.

2. Methodology

Presentation of the methodology is structured into three Sections. The properties of the main building elements and the mathematical programming representation are presented in Sections 2.1 and 2.2, respectively. Section 2.3 explains the autonomous sequencing of the ASF parametrization.

2.1. Properties of the building elements

Here, we either introduce or summarize the concepts, providing the building elements of the methodology. The CAF and the Pareto solutions' representation are discussed in Sections 2.1.1 and 2.1.2, respectively. Sections 2.1.3 explains the concept of extreme points of the

Pareto set, called corners. The space of achievements and the representation of the Pareto Front are discussed in Sections 2.1.4 and 2.1.5, respectively. The role of cuboids in our approach is explained in Section 2.1.6. Finally, Section 2.1.7 defines the applied measures of the PFR characteristics.

2.1.1. Criterion Achievement Function

In order to treat all criteria equitably, we adapt the concept of criteria achievements established by the A/R-method. This approach measures criteria performance in a common achievement scale. It is implemented by the Criterion Achievement Function (CAF):

$$caf_i(q_i, a_i, r_i) = PWL_i(q_i, a_i, r_i), \quad i \in I, \tag{6}$$

where $i \in I$ indexes criteria, $PWL(\cdot)$ stands for a Piece-Wise Linear function, strictly decreasing/increasing for minimized/maximized criterion, respectively, q_i stands for the core-model variable representing i-th criterion, and a_i, r_i are aspiration/reservation values, respectively. The $\boldsymbol{a}$ and $\boldsymbol{r}$ in the interactive MCMA define the criteria values one wants to achieve/avoid, respectively.

The $PWL(\cdot)$ used in (6) is parametrized for each criterion in order to transform the criteria values into the common scale [0, σ]. Therefore, the $PWL_i(\cdot)$ has the property:

$$PWL_i(u_i, u_i, n_i) = \sigma \text{ and } PWL_i(n_i, u_i, n_i) = 0, \quad i \in I, \tag{7}$$

where σ is a given scale, and u_i, n_i are the values of the Utopia and Nadir components, i.e., the best and the worst (within the Pareto set) criteria values, correspondingly.

Moreover, due to the desired characteristics, the $PWL(\cdot)$ is concave. Therefore, it can be represented by a set of auxiliary variables and linear constraints. We don't present here such a representation as it is commonly known; it can be found e.g., in Granat and Makowski (2000).

Following the popular achievement scaling, the `pyMCMA` implementation uses $\sigma = 100$. Therefore, the results presented in this paper also use this scale.

We summarize the key properties of the CAF:

- The criteria performance is interpreted in terms of achievements, i.e., is scaled to the range [0, 100], corresponding to the worst and best values within the Pareto set, respectively.
- CAF enables an easy comparison of the performance of criteria, each defined by the corresponding core-model variable in diverse measurement units, and typically having values of diverse orders of magnitude.
- CAF provides a uniform treatment of minimized and maximized criteria by measuring the performance of each criterion in terms of the corresponding achievement.

Broader discussion on the CAF, including its relations to the Membership Function of Fuzzy Sets, as well as on the representation of the PWL function in mathematical programming terms, is available in Granat and Makowski (2000). The discussion and examples of using this approach for the analysis of simultaneously achievable goals for conflicting criteria can be found in Makowski (2009).

2.1.2. Pareto solutions

Each solution in the presented approach is defined by the values of all variables in the integrated model defined in Section 2.2. However, for brevity sake of the presentation, we discuss below the solutions only in terms of the criteria achievements. Therefore, consider a set of solutions P:

$$\boldsymbol{p}_k = (vcaf_{ik},\ i \in I),\ k \in P \tag{8}$$

where $k \in P$ indexes the members of the current set (for each iteration) P, and $vcaf_{ik}$ denote the values of the CAF $caf_{ik}(\cdot)$ specified at the k-th iteration for i-th criterion. We recall that such solutions are feasible for the core model.

For brevity, in the sequel, when referring to the k-th Pareto solution, we often use the term Pareto point, or simply a point, denoted by $\boldsymbol{p}_k$, components of which are defined by (8).

2.1.3. Pareto Front corners

Corners of the PF are defined by its extreme points. Each corner is computed by optimizing the corresponding criterion. The prevailing approach to corners' computation is through the so-called selfish optimizations, i.e., optimizing in row each criterion with a regularizing term to avoid weakly Pareto solutions. However, such an oversimplified way of computing the PF corners determines only one, out of possibly an infinite number of, solution that maximizes the achievement of the corresponding criterion. Such situations occur in practice; see Appendix A for an example of a real model having an infinite number of solutions maximizing one of the three analyzed criteria. Generally, it is practically impossible to determine the uniqueness of the selfish optimization results from the analysis of a core model and the corresponding criteria specification. Therefore, in order to properly define the corners of the PF, we implemented a dedicated algorithm for this purpose.

For the dedicated regularized optimization aimed at computing the PF corners we define for each criterion $i \in I$ a set of auxiliary functions $corner_{ik}(\cdot)$:

$$corner_{ik}(\boldsymbol{caf}) = caf_i(q_i) + \epsilon \cdot caf_k(q_k) + \frac{\eta}{n-1} \cdot \sum_{j \in I \setminus \{i,k\}} caf_j(q_j)), \quad \forall\ [i,k] \in I,\ k \neq i, \tag{9}$$

where functions $caf(\cdot)$ are the CAF of the corresponding criterion defined by (6), parameters $\epsilon \gg \eta$ are small numbers serving as regularizing coefficients, n stands for the number of criteria, and $I \setminus \{i,k\}$ denotes the set I with indices i and k removed.

Function (9) is similar to a typical regularized single-criterion optimization objective. The main features of (9) are as follows:

- All terms of the function are in the common scale provided by the corresponding feature of the $caf(\cdot)$.
- Two regularizing coefficients ϵ, η allow for splitting the regularizing term into two terms in order to appropriately treat possibly conflicting criteria.
- The parameters (a_i, r_i) of the function $caf(\cdot)$ are defined differently for the first term and for the last two terms of (9); namely, in terms of the achievements, by $(100, 67)$ and by $(33, 0)$, respectively. The combination of (ϵ, η) and such values of (a_i, r_i) assures:
  - ⋆ Each solution to be a candidate ϵ-properly Pareto solution corresponding to a PF corner.

⋆ A good approximation of the achievement's values in the candidate corner of all but i-th criteria.

• The function (9) is used for each criterion iteratively to examine all combinations of criteria.

Finally, we explain the meaning of the "candidate" used above. Maximization of (9) sometimes results in providing a dominated solution. Therefore, the iterative procedure for corner computations includes checking the dominance relation and filtering out dominated solutions.

2.1.4. Space of achievements

The problem formulation presented in Section 1.3.2 deals with the Pareto solutions in the corresponding $\mathbb{R}^k$ space. All Pareto solutions are included in the corresponding hyper-cuboid defined by the U/N (Utopia and Nadir) points. The corresponding criteria values are in diverse measurement units and usually differ, for real-life problems, by many orders of magnitude.

In order to equitably treat all criteria, we evaluate the PFR characteristics in the achievement space, which is a linear transformation (by the CAFs) of the corresponding $\mathbb{R}^k$ criteria space. Therefore, the criteria achievements belong to the hyper-cube of the size σ. Furthermore, the achievements of the Pareto solutions are included in the hyper-cuboid defined by the Pareto Front corners; the latter is a part of the hyper-cube containing all Pareto solutions.

We note that the optimization is done in the model variable space, except that the pairs defining A/R are selected based on the criteria achievements in such a way that the desired PFR is reached. Then the A/R values in the achievement space are transformed into the corresponding values in the model variable space.

2.1.5. Pareto Front Representation

The PFR consists of the set of Pareto solutions denoted by $\boldsymbol{p} \in P$. The set P is initiated with the PF corners and then is extended iteratively. The distribution of the PFR is evaluated by the gap, defined as the L^∞ distance (aka the Chebyshev distance) between the pair of the currently most distant neighbor solutions. Each iteration provides one solution that decreases the PF gap. The iterations continue until the gap is reduced below the given parameter value.

The use of the Chebyshev norm is consistent with the adopted A/R method, which in turn was motivated by Rawls' theory of justice Rawls (1973). Moreover, the choice is also justified by the requirement of equitable criteria treatment, which implies a preference for improvement of the currently weakest achievement. However, we point out that, due to the property of the applied method, each iteration decreases not only the gap but also the distance measured by the L^1 norm, aka the Manhattan distance.

2.1.6. Cuboids

The commonly used in the context of Pareto efficient solutions concept of cuboids (formally hyper-cuboids) combined with the adaptation of the A/R-method for autonomous generation of the PF provides the basic framework for the presented method.

Let us consider the cuboid virtually defined by a pair of two neighbor Pareto points, say $(\boldsymbol{p_m}, \boldsymbol{p_n})$, see Fig. 1. The neighbor means that there is no already found Pareto solution located in such a cuboid. Ogryczak (2026) provides a formal definition of such neighbors.

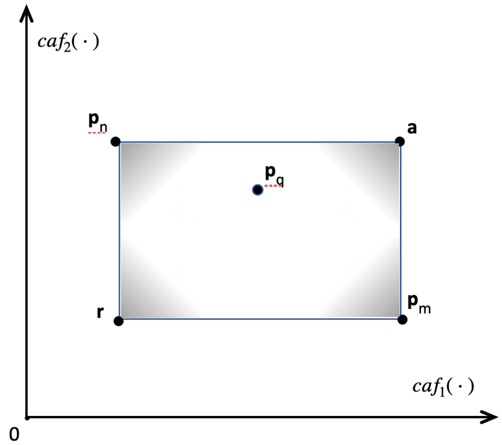


Figure 1: Cuboid defined by $(\boldsymbol{p_m}, \boldsymbol{p_n})$.

The considered cuboid contains many solutions, each of which is either efficient or dominated or infeasible. Let us consider a new (yet unknown) Pareto solution denoted by $\boldsymbol{p_q}$ and shown in Fig. 1 although its location still needs to be determined. Obviously, irrespective of the actual location of $\boldsymbol{p_q}$ in the cuboid, the following relations hold:

$$\|\boldsymbol{p_q} - \boldsymbol{p_m}\|_{L^\infty} < \|\boldsymbol{p_n} - \boldsymbol{p_m}\|_{L^\infty} \text{ and } \|\boldsymbol{p_q} - \boldsymbol{p_n}\|_{\mathrm{L}^\infty} < \|\boldsymbol{p_n} - \boldsymbol{p_m}\|_{\mathrm{L}^\infty}. \tag{10}$$

In other words, the new solution will decrease the gap defined by the pair of solutions $(\boldsymbol{p}_n, \boldsymbol{p}_m)$ defining the cuboid.

We now explain how a Pareto point $\boldsymbol{p_q}$ is found at each iteration. Using the values of the achievements of solutions $\boldsymbol{p_m}$ and $\boldsymbol{p_n}$ defining the cuboid, we consider two other cuboid vertices denoted $\boldsymbol{a}$ and $\boldsymbol{r}$, respectively. To illustrate the approach, we consider a bi-criteria problem; thus the components of $\boldsymbol{a}$ and $\boldsymbol{r}$ are defined by:

$$a_1 = \max(vcaf_{1m}, vcaf_{1n}),\ a_2 = \max(vcaf_{2m}, vcaf_{2n}), \tag{11}$$

$$r_1 = \min(vcaf_{1m}, vcaf_{1n}),\ r_2 = \min(vcaf_{2m}, vcaf_{2n}). \tag{12}$$

Note that the operators `max` and `min` are applied to select (from solutions defining the cuboid) for each criterion the better and the worse achievement, respectively. We recall that the better/worse is represented by the corresponding larger/smaller CAF value, and thus it is independent of the criterion (max- or minimized) type.

In other words, the components of $\boldsymbol{a}$ and $\boldsymbol{r}$ are defined for each criterion by the corresponding achievements of the solutions defining the cuboids, best and worst, respectively. This, in turn, obviously implies that $\boldsymbol{r} \prec \boldsymbol{v}$, where $\boldsymbol{v}$ denotes any Pareto solution lying within the cuboid. Conversely, $\boldsymbol{v} \prec \boldsymbol{a}$, i.e., $\boldsymbol{a}$ is not attainable; the latter is also implied by the property of the Pareto solutions defining the cuboid.

Next, we recall the definition of the Achievement Scalarizing Function $ASF(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{r})$ applied in numerous interactive MCMA:

$$ASF(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{r}) = \min_{i \in I} caf_i(q_i,\ a_i,\ r_i) + \frac{\epsilon}{n} \cdot \sum_{i \in I} caf_i(q_i,\ a_i,\ r_i), \tag{13}$$

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, and ϵ is a small positive parameter. Finally, the Pareto solution that best fits the preferences specified by $(\boldsymbol{a},\ \boldsymbol{r})$ can be found by solving the following optimization problem:

$$\hat{\boldsymbol{x}} \ = \ \arg\max_{x \in X} \ ASF(\cdot), \tag{14}$$

where X is the set of feasible solutions of the core model (implicitly defined by the relationship of the model). The structure of the mathematical programming model that integrates the core model with the implementation of $ASF(\cdot)$ is presented in Section 2.2. Here, we only point out that: (1) the representation of the ASF in the integrated model does not shrink the set X, and (2) optimization of (14) always provides an ϵ-properly Pareto solution. A detailed discussion of the underlying methodology can be found in Wierzbicki et al. (2000) and several earlier publications cited therein.

We recall that solution $\hat{\boldsymbol{x}}$ contains the criteria values; therefore, the corresponding Pareto point, denoted by $\boldsymbol{p}_q$ and marked in Fig. 1, will be added to the current PFR, namely set P. Moreover, due to the properties of $(\boldsymbol{a},\ \boldsymbol{r})$, the newly found $\boldsymbol{p}_q$ will be somewhere within the cuboid defined above. The new solution will be considered, together with all previously computed solutions, including each component of the $(\boldsymbol{p}_m, \boldsymbol{p}_n)$ pair, to define new cuboids. The pair $(\boldsymbol{p}_m, \boldsymbol{p}_n)$ will no longer be neighbor solutions; therefore, this pair will not be used again for a cuboid definition. Note that if $(\boldsymbol{p}_m, \boldsymbol{p}_n)$ defines a possible irreducible gap (i.e., are the most distant neighbor solutions) in the Pareto set, then $\boldsymbol{p}_q$ will be at either $\boldsymbol{p}_m$ or $\boldsymbol{p}_n$. We point out that such irreducible gaps cannot occur in the MCMA of LP core models, but are likely in the analysis of MILP or NLP models.

2.1.7. PFR characteristics

We consider the characteristics in the criteria space transformed through CAFs to the metrics common for all criteria. For the scaling selected in Sec. 2.1.1 all Pareto solutions are contained in the hypercube of size σ. Moreover, they are actually contained in the cuboid defined by the corners defined in Sec. 2.1.3.

We use the following measures of the quality of the PFR evaluated in the achievements' space defined in Section 2.1.4:

- gap γ is defined by

$$\gamma = \max_{\boldsymbol{p1} \in P}\big(\min_{\boldsymbol{p2} \in P \ | \ \boldsymbol{p1} \neq \boldsymbol{p2}} \|\boldsymbol{p1} - \boldsymbol{p2}\|_{L^\infty}\big), \tag{15}$$

  where $\boldsymbol{p1}, \boldsymbol{p2}$ is a pair of neighbor solutions.
- accuracy $\alpha = \max(\gamma - \rho, 0)$, where the parameter $\rho > 0$ is the requested accuracy of the PFR.
- cardinality κ - number of the PFR elements.

Additionally, the presented approach guarantees that every new Pareto solution improves the PFR characteristics defined in Sec. 2.1.7. This property, formally represented by the relation (10), is implied by the location of each new solution inside the corresponding, sequentially generated earlier, cuboids having vertices defined by pairs of neighbor points

(Pareto solutions) distant by more than the required resolution of the PFR. Furthermore, we note that a solution found by using the $(\boldsymbol{r}, \boldsymbol{a})$ pair is rarely located in the line connecting these points because the above defined $ASF(\cdot)$ is a non-linear function.

2.2. Mathematical programming representation

name spaces of variables	mc-model	core-model
integrated model variables	$\boldsymbol{v}$ $\boldsymbol{q}$	$\boldsymbol{q}$ $\boldsymbol{x}$
mc-model variables	$\boldsymbol{v}$ $\boldsymbol{q}$	
core-model variables		$\boldsymbol{q}$ $\boldsymbol{x}$
mc-model relations	**M**	0
mc-and core-model link	**I**	**-I**
core-model relations	0	**C**

Figure 2: Jacobian structure of integrated mc- and core-models.

Fig. 2 shows the structure of the Jacobian of the linked models. In terms of mathematical programming, the mc- and core-models are implemented in separate name-spaces to avoid conflicts of possible same variables' names. Moreover, the pairs of variables $\boldsymbol{q}$ representing criteria in the corresponding name spaces have the same name. The two submodels are linked by simple equality relations that involve such pairs. Other variables of each model are shown in Fig. 2 named by $\boldsymbol{v}$ and $\boldsymbol{x}$, respectively. The mc- and core-model's relations are represented by matrices $\boldsymbol{M}$ and $\boldsymbol{C}$, respectively.

The mc-model is a small LP model. Its symbolic specification (aka abstract model) has only several compound variables and relations; it is invariant of the core-model representation, except of the set of names representing criteria. The latter names are specified in the configuration file; this enables the use of `pyMCMA` software without modifications for various core models. The model instance (aka concrete model) is composed of several dozens of continuous variables and linear relations; it is generated for each iteration as its A/R parameters are different at each iteration.

As the mc-model is a small LP model, the optimization of the linked models is typically run by the same solver as used for the core model development; also the computation resources are similar for MCMA of non-trivial core models.

Implementing the linkage is very easy for the models implemented in Pyomo, the open source Python-based optimization modeling language (Bynum et al., 2021), which supports easy seamless integration of concrete models. Therefore, this Pyomo feature enables their seamless integration and thus re-usability of the `pyMCMA` open-source package for any LP

(including MILP) core model. The details of the implementation are described in Makowski et al. (2024).
One should also note that problems having large ranges of criteria values pose numerical challenges. Therefore, a proper implementation of (6) requires adaptive scaling, which in turn requires the corresponding modeling and programming skills. Therefore, the developed approach provides an effective implementation of the *mc-model* and its seamless integration with the analyzed *core-model*. In this way, it removes a key obstacle in a wide MCMA use. The `pyMCMA` addresses one of the main obstacles in the effective use of MCMA, i.e., implementation of a representation of parameterized user preferences in a form required by optimization software. The latter actually requires a submodel seamlessly integrated with the core model and represented by an optimization task. This approach has been implemented by the `pyMCMA` authors and successfully applied in interactive MCMA of diverse problems. Proper implementation of an MCMA submodel requires both expertise and resources; therefore, it is rarely done. Thus, `pyMCMA` fills the functionality gap in MCMA methods and reusable tools.

2.3. Sequential generation of cuboids

The rules governing the sequence of cuboids' generation are the key issue for combining the computational efficiency with reaching the desired characteristics, defined in Section 2.1.7 of the generated PFR. We present two methods of generating the cuboids, named WMO[2] and SPFR, discussed in Sections 2.3.1 and 2.3.2, respectively. Section 3 compares the results achieved by using these methods.

2.3.1. WMO

Our original WMO approach assumed the generation of all possible cuboids for all pairs of sequentially found Pareto points (excluding those close to any previously found solution) and then selecting, for each next iteration, the largest cuboid (i.e., the one generated by the most distant pair of the Pareto points) after checking that the cuboid does not include any previously found solution. The results of this approach's application are reported in Makowski et al. (2024, 2025) show that the approach is correct in the sense of providing the PFR of the desired resolution. However, we noticed that the number of generated cuboids was growing very fast with decreasing the requested accuracy α. Moreover, solutions lying close to the PF boundary were missed for some models.
It was obvious, that generation of all possible cuboids and checking if they are empty, takes more and more time as the number of generated solutions grows. We have tried diverse ways to speed up these processes, but none of them were satisfactory. The lessons from those efforts led us to develop the new approach described below.

[2]The WMO acronym corresponds to the initials of the late Professor Włodek Ogryczak of the Warsaw University of Technology, who developed the original WMO approach, see Ogryczak (2026).

2.3.2. SPFR

The new approach is based on structured sequential generation of neighbor solutions assuring: (1) achievement of as close as reasonably possible uniformly distributed points representing the PF, (2) covering the whole PF with the desired accuracy, and (3) computational efficiency. The approach is composed of two stages:

1. Generation of the PF envelope, see examples in Section 3.1. The envelope consists of solutions on parts of the PF boundary, each generated by the corresponding pair of the PF corners. Solutions are added to each part as long as the gap between neighbor points is reduced below the requested resolution. Therefore, the envelope can be considered as a PFR. This, however, is not a desired representation because it does not include solutions lying between the points belonging to different parts of the envelope.
2. Filling the PF envelope, see examples in Section 3.2. We apply here a similar approach to that of the first stage. Namely, we select pairs of solutions from each pair of the parts of the envelope and treat them in the same way as corner solutions in the first stage.

Due to the similarity of the processes of the two stages, for brevity sake, we use the term edge for the above-defined segments of the PF boundary, as well as for the pairs of points defining the edges in the second stage. Therefore, each edge is initiated by two points (Pareto solutions), which are further termed the anchors. Thus, in the first stage, the anchors are defined by the PF corners, and in the second stage the anchors are defined by pairs of points lying in the selected pairs of edges (originating from the same anchor) generated in the first stage.

3. Case studies

We illustrate the PFR generation using the following four core models:

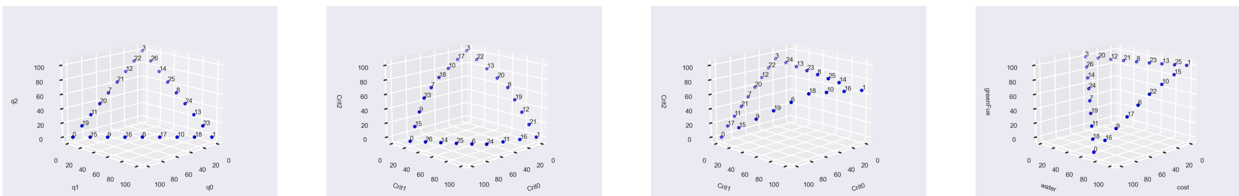
- `Alves`: the model by Alves and Costa (2009) illustrating the difficulty of finding the Nadir. This model shows that the algorithm presented in this paper not only generates a representation of the Pareto frontier, but also correctly computes the Nadir point.
- `Benson`: the commonly known model of Benson and Sayin (1997). We use this model to show the properties of our approach because the shape of the PF of this model is known from several publications.
- `Pipa`: the model of technology pathways of China's liquid fuel production, see e.g., Zhao et al. (2022); Ding et al. (2022, 2023). The detailed discussion of the MCMA of this model is available in Makowski et al. (2024, 2025).
- `Plain`: a very simple model, defined by a simple relation:

$$\sum_{j \in J} x_j \leq 1, \quad J = \{0, 1, \ldots, 4\}. \tag{16}$$

  One can easily make the MCMA of this model for any number of criteria less than six. The shape of the corresponding PF is obvious; therefore, the model primarily serves testing purposes.

The `Pipa` model has the documented specification; therefore, the results of its analysis have a concrete interpretation in terms of not only the criteria achievements but also of the trajectories of the decisions, outcomes, as well as diverse auxiliary variables. The interpretation of the results of the `Plain` is obvious. However, there is no interpretation of the trade-offs between criteria achievements because these models are abstract.
The PFR generated by the SPFR method for each of these models is shown below for three criteria. Section 3.3 discusses the comparison of generation by the WMO and SPFR methods.

3.1. Pareto Front envelope



(a) Plain model. (b) Benson model. (c) Alves model. (d) Pipa model.

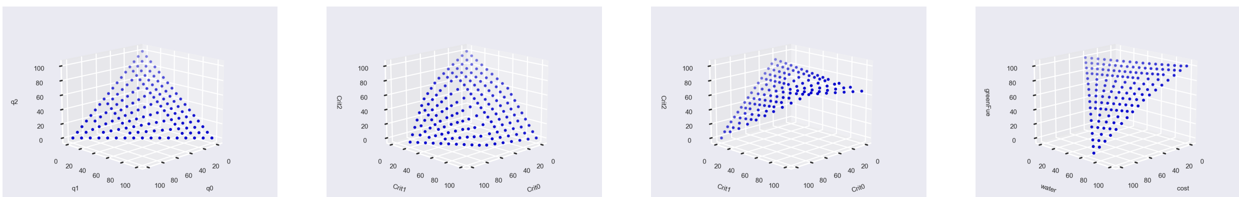
Figure 3: Envelopes of the Plain, Benson, Pipa models, for $\alpha = 20$.

The envelope of the PFR of each model analyzed with three criteria consists of three edges. Each edge is initialized by the pair of anchors defined by the corresponding PF corners. The corners of the PF are labeled by 0, 1, and 3, respectively. To show the sequence in which the points (Pareto solutions) are generated, Fig. 3 shows, for each model, the envelopes generated for a rather large value of the requested resolution, namely $\rho = 20$. The markers of each point show the point generation sequence.
We point out that all solutions in the edge defined by the pair corners labeled 1 and 3 in Fig. 3d have the same value of the criterion greenFuel. This illustrates the issues of non-unique corners discussed in Section 2.1.3.
We briefly introduce the procedure for specifying pairs of points used for subsequent specification of cuboids, each serving the computation of the corresponding Pareto solution. The procedure consists of two phases.
In the first phase, the already computed points, separately in each edge, are sorted by the distance from an anchor of this edge. Then, for each neighbor points the distances between them are computed, and the pairs of points that are distant by more than the required resolution ρ are stored in the list of candidate pairs. The list is common for all edges. Next, the list is sorted by the distances in descending order, and it is ready for the next phase. If the newly created list is empty, then the envelope computation is complete and the next stage (filling the envelope, Sec. 3.2) starts.
The second phase of computing the PF envelope consists of iterations. At the beginning of each iteration, the dedicated interface function is called by the workflow. The called function temporarily stores the pair from the list top, removes the pair from the list, and returns the pair to be used for defining the next cuboid. After the next solution is found, the function is called again. If the list of candidate pairs is empty, then the procedure returns to the first phase.

We notice the computation's efficiency of the above summarized procedure. The candidate list is prepared for all neighbor-pairs in all edges because: (1) the edges can be filled independently, and (2) each neighbor pair included in the list defines a cuboid, in which the corresponding new solution is located; therefore, all (new and previously computed) solutions can be considered after the list of candidate pairs has been processed anew.

3.2. Filling the envelope

The solutions that fill the envelope are generated using the same procedure as the solutions that define the envelope described in Section 3.1. The resulting PFR of the three models, each for the $\rho = 10$, are shown in Fig. 4.



(a) Plain model. (b) Benson model. (c) Alves model. (d) Pipa model.

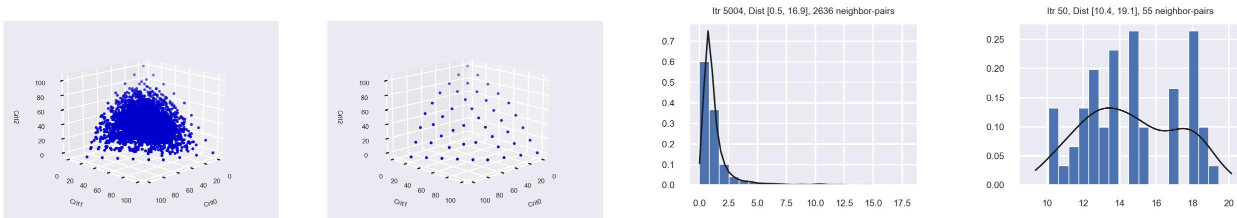
Figure 4: Pareto Front representations for $\rho = 10$.

The only difference between the two stages of generating the PFR is the way in which the anchors of the corresponding edges are defined. The anchors of the envelope's edges are defined by the PF corners. The anchors of second stage edges (lying within the envelope) are defined by pairs of solutions selected from the envelope's edges, excluding their anchors. The procedure of defining the second stage's edges is rather simple. First, we note that for three criteria analyses, one needs to consider only two edges of the envelope originating from the common solution, further on, referred to as a base. Solutions in each of these two edges are sorted, separately for each edge, by the distance from the base. Next, the pairs of consecutive points (starting from the closest to the base) serve as anchors of the edges lying within the envelope; the latter edges are called internal edges. Then, the solutions lying in each internal edge are processed in the same way as described in Section 3.1. Namely, pairs of points defining cuboids are selected from each internal edge, and new solutions are added to the edge to which the points defining the corresponding cuboid belong. Similarly to the first stage, the candidate list is common for all internal edges.

3.3. Comparison of the WMO and SPFR methods

We briefly compare the two iterative methods of the PFR generation:

1. the original WMO method that assumes generation of all possible cuboids defined by the current PFR, and selection for each iteration the largest empty cuboid. The method's description and its application to MCMA of the Pipa model is available in (Makowski et al., 2024).
2. the newly developed and presented SPFR method presented in Section 2.3.2.



(a) WMO method. (b) SPFR method. (c) Distances of PFR 5a. (d) Distances of PFR 5b.

Figure 5: Comparison of unstructured and structured generation of the Pareto Front Representations of the Benson model for $\rho = 20$.

Fig. 5 shows the comparison of these two methods applied for generation of the PFR of the Benson model for $\rho = 20$. Figs. 5a and 5b show the PFR obtained by the unstructured and structured method, respectively. Figs. 5c and 5d show the the corresponding distributions of distances between neighbor solutions.

Obtaining the PFR even for the rather coarse $\rho = 20$ for the Benson model requires much more iterations than for the Pipa model, analysis of which is presented in (Makowski et al., 2024). For the unstructured method we run 50,000 iterations and still there are several neighbor solutions distant by more then 20 (the largest gap γ, as shown in Fig. 5c is equal to 38.7). Moreover, almost all computed solutions were filtered out because they were too close to another solution. Therefore, Fig. 5a shows the remaining 541, out of 50,000 computed, solutions. Fig. 5c also shows that the majority of the PFR solutions are close to each other; the distances between neighbors larger than 10 are not seen because their number is too small to be shown in the Figure.

The distribution of distances between neighbor solutions is shown in Fig. 5d shows the clear advantage of the structured generation of the PFR illustrated by Fig. 5b and obtained by only 50 iterations.

Moreover, the results of the structured generation of the PFR shown in Section 3.2 confirm that this method is very efficient also for other core-models.

In order to characterize the features of the presented methods, we summarize in Table 1 the key characteristics of the PFR computations for the four discussed models and the diverse requested resolutions ρ. The columns κ and `iter` show the numbers of the PFR's solutions and the iterations, respectively. Column `time` shows the wall-clock time (in seconds) of the `pyMCMA` open-source package run on the iMac M1, 16GB RAM, macOS Sequoia 15.6.1 using the open-source GLPK optimization solver.

The first group of four runs in Table 1 shows that the structured generation of the PFR for each of the four models for the same requested ρ is very efficient. We point out that the first three models are tiny while the `Pipa` model is substantially larger (it has 460 variables and 430 relations). Therefore, the core-model sizes explain the different computation times.

Table 1: Characteristics of computations.

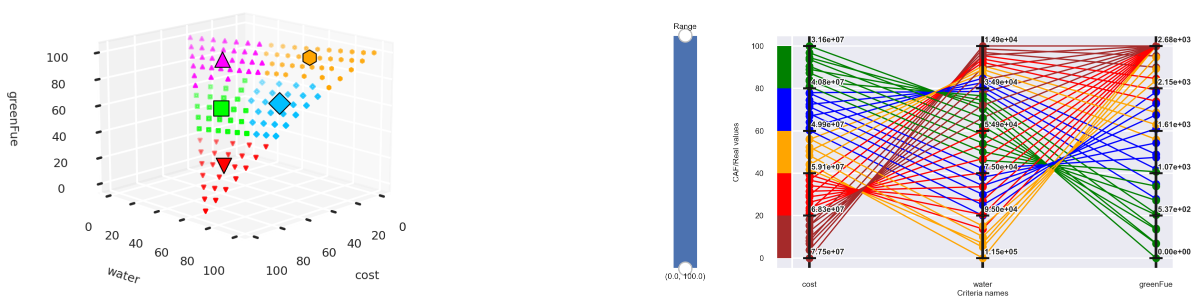
Model	ρ	κ	`iter`	time [s]
plain	10	146	149	2.3
benson	10	166	168	2.6
alves	10	129	132	2.1
pipa	10	142	145	3.6
pipa	3	1269	1271	29.1
benson	20	48	50	1.2
benson*	20	2637	5005	125.2

Generation of the PFR for the $\rho = 3$ obviously takes about 8 times longer than for $\rho = 10$, but it is still fairly fast. Finally, we point out that the PFR cardinality, represented by the corresponding number of Pareto solutions, depends on the requested resolution ρ.
The last two runs compare the structured and the unstructured PFR generations (the latter marked by benson*). The unstructured generation not only took about 100 times longer, but the results shown in Fig. 5 show that the corresponding PFR is obviously worse than that received by the structured generation. Moreover, we point out that using the unstructured method for small ρ (i.e., high resolution of the PFR) requires prohibitively long computations. In most cases of structured generation, the number of solutions in the PFR is only slightly smaller than the number of the iterations; this indicates a tiny number of solutions close to each other that are pruned on the fly. However, in the unstructured generation, almost half of the generated solutions were pruned on the fly.

4. Posterior analysis of the PFR

The PFR is composed of many (thousands for a high requested resolution, i.e., small values of ρ) Pareto solutions. Therefore, we support its analysis by visualization of the basic PF properties. For illustration, we show here only two plots:

- Clusters of the Pareto solutions shown in Fig 6a. The default number of five clusters can optionally be changed.
- Trade-offs between the criteria achievements illustrated by the parallel coordinates plot in Fig 6b.



(a) PFR clusters.

(b) Criteria achievements.

Figure 6: PFR clusters and trade-offs between criteria achievements. The markers of the clusters correspond to the medoids, i.e., the solutions closest to the center of the corresponding cluster.

The pyMCMA package displays (during the computations) several more plots, including projections of the criteria achievements on subspaces corresponding to the criteria pairs, distributions of the Pareto solutions. All plots are stored; therefore, they are available for later reviews.

The PFR's analysis is typically augmented by the analysis focused on the needs for which the core model is developed. Therefore, the `pyMCMA` application stores also the data related to each Pareto solution. The data includes the values of criteria (in the model measurement units and in the achievement scale) as well as values of the model variables optionally selected by the users.
More information about the stored data is available in the online `pyMCMA` documentation described in Makowski et al. (2025).

5. Main features of the methodology

We summarize here the main features of the presented methodology:
1. It autonomously (i.e., without parametrization) generates PFR for any LP model, provided that the model defines variables that can be used as criteria. There are no other requirements for independently developed LP models.
2. Neither programming nor multiple-criteria analysis knowledge is required for using the provided open-source application implementing the methodology. The provided LP model is seamlessly aggregated with the provided `pyMCMA` application. The users only need to specify, in a simple text file, the criteria and the model location in the corresponding local file system. Therefore, generation of the PFR can be done as easily as running any LP solver.
3. All criteria are treated equitably (objectively, preference-free) in the whole set of Pareto solutions, which is a requirement for scientific problem analysis. There are no requirements about criteria mutual dependencies. Moreover, the provided results also support preference-oriented explorations of diverse Regions of Interest (ROI).
4. The method works for any shape of the (unknown) PF. The minimized and maximized criteria are handled uniformly. Other criteria types, e.g., target, related to trajectories, state-variables, etc., are also easy to implement. The method is invariant of the number of criteria, although it has far been intensively tested for three criteria. Moreover, the method is numerically robust as it handles also models having criteria ranges that differ by many orders of magnitude. This is achieved by internal adaptive scaling of the applied ASF.
5. The designed structure of the neighbor solutions increases computation's efficiency and results in the desired distribution of the PFR. Each iteration generates one efficient solution that decreases the maximum gap of the current PFR.
6. Posterior analysis is supported by visualization of results by 2D and 3D graphs, as well as by a parallel coordinates plot showing the criteria trade-offs. Moreover, clustering of the PFR elements shows a small number of medoids representing the corresponding subsets of the PF. Additionally, results of all iterations are optionally exported for problem-specific analysis in the space of the model variables.

We close the method summary by showing in Fig. 7 the computation flows between the main elements of the method. It starts with the selected `pipa` LP model and the corresponding `cfg.yml` file specifying the model location and the selected variables defining criteria. The selected criteria are used for the parametrization of the LP model representing the ASF; this model is seamlessly linked with the analyzed model, forming the combined LP model,

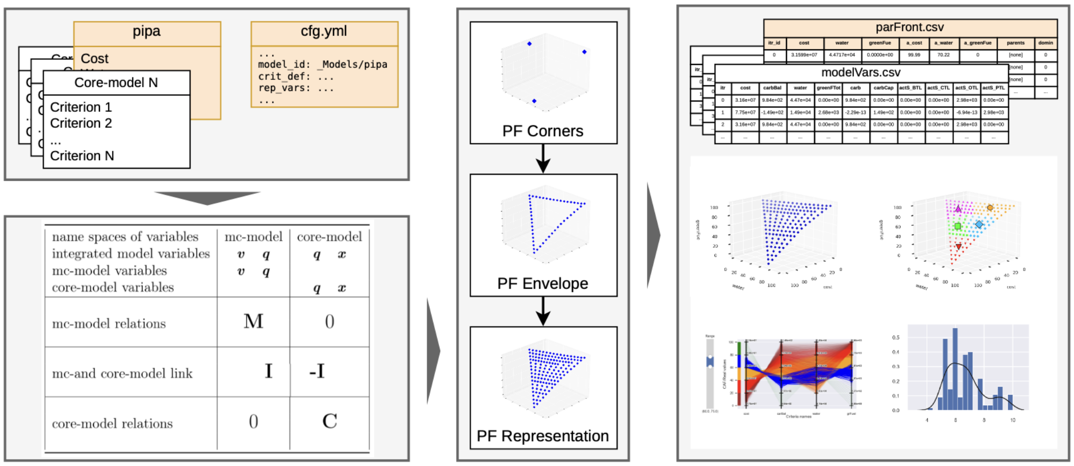


Figure 7: Basic elements of method.

optimization of which provides elements of the PFR. The middle block shows the stages of the structured generation of the PFR. The right block illustrates selected elements supporting the posterior analysis.

6. Conclusions

The presented methodology and its implementation in the open-source `pyMCMA` package effectively provide an autonomously generated representation of the PF. The method's main features, summarized in Section 5 show the effective support of a scientific project aiming at providing an objective problem analysis with consideration of several criteria. The structured generation of the PFR dramatically improved (as compared with the original method) the computational efficiency, as well as the desired characteristics of the PFR. The method does not require any parametrization; therefore, it can be used in similar ways as commonly used optimization solvers. This, in turn, shall enable a broader MCMA application to complex real-world problems analyzed with mathematical programming models, which continue to be hindered by the lack of modular, easy-to-use MCMA tools. In contrast to iterative approaches that require preference specification at each iteration, the method offers autonomous generation of PFR covering the entire PF.

Acknowledgments

The presented methodology is based on the approach proposed by late Professor Włodek Ogryczak in discussions with the authors of this paper in 2009-2010. Ogryczak (2026) contains the original proposal written by Professor Ogryczak.

Appendix A. Non-unique corner solutions

In order to briefly illustrate the issue of a non-unique selfish optimization solution, we consider the analysis of the Pipa model with three criteria (cost, water, grFuel). Such a model instance has an infinite number of solutions maximizing the grFuel criterion; two of these solutions define two PF corners, other solutions lie on the edge connecting these corners.

Table A.2: PF corners of the Pipa model.

Corner ID	Cost	Water	grFuel
OTL	100	70	0
BTL	44	0	100
PTL	0	100	100

Table A.2 shows the criteria achievements of the corner solutions. Corner IDs denote the only technology used in the corresponding optimal solution. The interpretation of the corner solutions is straightforward, when one considers the model specification: the cheapest solution (marked by the OTL) uses a moderate amount of water and does not produce any grFuel. The maximum amount of grFuel can be produced by diverse combinations of two technologies (known as BTL and PTL, respectively), each of such a combination being characterized by different costs and water use. We also point out that there is an infinite number of Pareto solutions in the edge defined by the BTL and PTL corners; thus, an oversimplified approach to the computation of the PF corners finds only one of infinitely many solutions that maximize the grFuel criterion.

References

Alves, M.J., Costa, J.P., 2009. An exact method for computing the nadir values in multiple objective linear programming. European Journal of Operational Research 198, 637–646. doi:10.1016/j.ejor.2008.10.003.

Antoine, J., Fischer, G., Makowski, M., 1997. Multiple criteria land use analysis. Applied Mathematics and Computation 83, 195–215.

Belton, V., Stewart, T., 2002. Multiple Criteria Decision Analysis. An Integrated Approach. Kluwer Academic Publishers, Boston.

Benson, H.P., Sayin, S., 1997. Towards finding global representations of the efficient set in multiple objective mathematical programming. Naval Research Logistics (NRL) 44, 47–67. doi:10.1002/(SICI)1520-6750(199702)44:1<47::AID-NAV3>3.0.CO;2-M.

Bynum, M.L., Hackebeil, G.A., Hart, W.E., Laird, C.D., Nicholson, B.L., Siirola, J.D., Watson, J.P., Woodruff, D.L., 2021. Pyomo–optimization modeling in Python. volume 67. Third ed., Springer Science & Business Media.

Chen, Y., Zou, X., Xie, W., 2011. Convergence of multi-objective evolutionary algorithms to a uniformly distributed representation of the pareto front. Information Sciences 181, 3336–3355. doi:10.1016/j.ins.2011.04.004.

Climaco, J. (Ed.), 1997. Multicriteria Analysis. Springer, Berlin.

Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V., 2007. Evolutionary algorithms for solving multi-objective problems. Springer.

Deb, K., Miettinen, K., Sharma, D., 2009. A hybrid integrated multi-objective optimization procedure for estimating nadir point, in: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.K., Sevaux, M. (Eds.), Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 569–583.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 6, 182–197.

Ding, B., Makowski, M., Nahorski, Z., Ren, H., Ma, T., 2022. Optimizing the technology pathway of China's liquid fuel production considering uncertain oil prices: A robust programming model. Energy Economics 115, 106371. Published online: October 21, 2022. DOI https://doi.org/10.1016/j.eneco.2022.106371.

Ding, B., Makowski, M., Zhao, J., Ren, H., Zakeri, B., Ma, T., 2023. Analysis of technology pathway of China's liquid fuel production with consideration of energy supply security and carbon price. Journal of Management Science and Engineering 8, 1–14. Published online: August 7, 2022. DOI https://doi.org/10.1016/j.jmse.2022.07.002.

Ehrgott, M., Köksalan, M., Kadziński, M., Deb, K., 2026. Fifty years of multi-objective optimization and decision-making: From mathematical programming to evolutionary computation. European Journal of Operational Research 330, 1–25. doi:10.1016/j.ejor.2025.06.012.

Eichfelder, G., 2009. Scalarizations for adaptively solving multi-objective optimization problems. Computational Optimization and Applications 44, 249–273. doi:10.1007/s10589-007-9155-4.

Faulkenberg, S.L., Wiecek, M.M., 2010. On the quality of discrete representations in multiple objective programming. Optimization and Engineering 11, 423–440.

Figueira, J., Greco, S., Ehrgott, M. (Eds.), 2005. Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, New York.

Fischer, G., Granat, J., Makowski, M., 1999. AEZWIN An Interactive Multiple-Criteria Analysis Tool for Land Resources Appraisal. World Soil Resources Reports 87. International Institute for Applied Systems Analysis, *and* Food and Agriculture Organization of the United Nations. Laxenburg, Austria, *and* Rome, Italy.

Geoffrion, A., 1968. Proper effiiciency and the theory of vector maximization. Journal of Mathematical Analysis and Applications 22, 618–630.

Geoffrion, A., 1987. An introduction to structured modeling. Management Science 33, 547–588.

Geoffrion, A., 1989a. Computer-based modeling environments. European Journal of Operational Research 41, 33–43.

Geoffrion, A., 1989b. Integrated modeling systems. Computer Science in Economics and Management 2, 3–15.

Geoffrion, A., Krishnan, R., 2001. Prospects for operations research in the e-business era. Interfaces 31, 6–36.

Granat, J., Makowski, M., 2000. Interactive Specification and Analysis of Aspiration-Based Preferences. European J. Oper. Res. 122, 469–485.

Grauer, M., Wierzbicki, A. (Eds.), 1984. Interactive Decision Analysis. volume 229 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, Berlin, New York.

Greco, S., Ehrgott, M., Figueira, J., 2016. Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, New York. doi:10.1007/978-1-4939-3094-4.

Isermann, H., Steuer, R.E., 1987. Computational experience concerning payoff tables and minimum criterion values over the efficient set. European Journal of Operational Research 33, 91–97.

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

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

Makowski, M., 2005. A structured modeling technology. European J. Oper. Res. 166, 615–648.

Makowski, M., 2009. Management of attainable tradeoffs between conflicting goals. Journal of Computers 4, 1033–1042. ISSN 1796-203X.

Makowski, M., Granat, J., Shekhovtsov, A., Nahorski, Z., Zhao, J., 2024. pyMCMA: Uniformly distributed Pareto-front representation. SoftwareX 27, 101801. doi:https://doi.org/10.1016/j.softx.2024.101801.

Makowski, M., Granat, J., Shekhovtsov, A., Nahorski, Z., Zhao, J., 2025. Version [2.0] — [pymcma: Uniformly distributed pareto-front representation]. SoftwareX 30, 102097. doi:https://doi.org/10.1016/j.softx.2025.102097.

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

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

Makowski, M., Wierzbicki, A., 2003. Modeling knowledge: Model-based decision support and soft computations, in: Yu, X., Kacprzyk, J. (Eds.), Applied Decision Support with Soft Computing. Springer-Verlag, Berlin, New York. volume 124 of *Series: Studies in Fuzziness and Soft Computing*, pp. 3–60. ISBN 3-540-02491-3, draft version available from http://www.iiasa.ac.at/~marek/pubs/prepub.html.

Márquez-Vega, L., Falcón-Cardona, J.G., Osuna, E.C., 2024. A multi-objective evolutionary algorithm based on uniformity and diversity to handle regular and irregular pareto front shapes. IEEE Access 12, 158878 – 158907. doi:10.1109/ACCESS.2024.3486255.

Mesquita-Cunha, M., Figueira, J.R., Barbosa-Póvoa, A.P., 2023. New ϵ-constraint methods for multi-objective integer linear programming: A Pareto front representation approach. European Journal of Operational Research 306, 286–307. doi:https://doi.org/10.1016/j.ejor.2022.07.044.

Messac, A., Ismail-Yahaya, A., Mattson, C., 2003. The normalized normal constraint method for generating the pareto frontier. Structural and Multidisciplinary Optimization 25, 86–98. doi:10.1007/s00158-002-0276-1.

Messac, A., Mattson, C.A., 2002. Generating well-distributed sets of Pareto points for engineering design using physical programming. Optimization and Engineering 3, 431–450.

Nakayama, H., 1994. Aspiration Level Approach to Interactive Multi-objective Programming and its Applications. Working Paper WP-94-112. International Institute for Applied Systems Analysis. Laxenburg, Austria.

Nakayama, H., Yu, Y., Yoon, M., 2009. Sequential Approximate Multiobjective Optimization using Computational Intelligence. Springer, Berlin, Heidelberg. ISBN 978-3-540-88909-0.

Ogryczak, W., 2026. Multiple efficient solutions generation. doi:10.5281/zenodo.19654373.

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

Pareto, V., 1896. Cours d'économie politique. volume 1. Librairie Droz, 1964.

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

Pascoletti, A., Serafini, P., 1984. Scalarizing vector optimization problems. Journal of Optimization Theory and Applications 42, 499–524. doi:10.1007/BF00934564.

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

Ren, H., Zhou, W., Makowski, M., Yan, H., Yu, Y., Ma, T., 2021. Incorporation of life cycle emissions and carbon price uncertainty into the supply chain network management of PVC production. Annals of Operations Research 300, 601–620. doi:10.1007/s10479-019-03365-1. published online: September 11, 2019.

Ren, H., Zhou, W., Makowski, M., Zhang, S., Yu, Y., Ma, T., 2023. A multi-criteria decision support model for adopting energy efficiency technologies in the iron and steel industry. Annals of Operations Research 325, 1111–1132. doi:10.1007/s10479-022-04548-z. published online: February 7, 2022.

Ruiz, F., Luque, M., Miguel, F., del Mar Muñoz, M., 2008. An additive achievement scalarizing function for multiobjective programming problems. European Journal of Operational Research 188, 683–694. URL: https://www.sciencedirect.com/science/article/pii/S0377221707004912, doi:https://doi.org/10.1016/j.ejor.2007.05.009.

Ruzika, S., Wiecek, M.M., 2005. Approximation methods in multiobjective programming. J. Optim. Theory Appl. 126, 473–501. doi:10.1007/s10957-005-5494-4.

Saini, B.S., Singh, H.K., Shavazipour, B., Miettinen, K., 2025. An efficient iterative approach for uniformly representing pareto fronts, in: Singh, H., Ray, T., Knowles, J., Li, X., Branke, J., Wang, B., Oyama, A. (Eds.), Evolutionary Multi-Criterion Optimization, Springer Nature Singapore, Singapore. pp. 241–256.

Sawaragi, Y., Nakayama, H., Tanino, T., 1985. Theory of Multiobjective Optimization. Academic Press, New York.

Sayın, S., 2000. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Mathematical Programming 87, 543–560. doi:10.1007/s101070050011.

Shao, L., Ehrgott, M., 2016. Discrete representation of non-dominated sets in multi-objective linear programming. European Journal of Operational Research 255, 687–698. doi:https://doi.org/10.1016/j.ejor.2016.05.001.

Steuer, R., 1986. Multiple Criteria Optimization: Theory, Computation, and Application. J. Wiley & Sons, New York.

Stewart, T., van den Honert, R. (Eds.), 1998. Trends in Multiple Criteria Decision Making. volume 465 of *Lecture Notes in Economics and Mathematical Systems*. Springer Verlag, Berlin, New York.

Utyuzhnikov, S.V., Fantini, P., Guenov, M.D., 2009. A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization. Journal of Computational and Applied Mathematics 223, 820–841.

Wang, H., Jin, Y., Yao, X., 2017. Diversity assessment in many-objective optimization. IEEE Transactions on Cybernetics 47, 1510–1522. doi:10.1109/TCYB.2016.2550502.

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

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

Wierzbicki, A., 1982. A mathematical basis for satisficing decision making. Mathematical Modelling 3, 391–405.

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

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

Yu, P., 1974. Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. Journal of Optimization Theory and Applications 14, 319–377. doi:10.1007/BF00932614.

Yu, P., 1985. Multiple-Criteria Decision Making: Concepts, Techniques, and Extensions. Plenum Press, New York, London.

Yun, Y., Nakayama, H., Arakawa, M., 2004. Multiple criteria decision making with generalized DEA and an aspiration level method. European Journal of Operational Research 158, 697–706. doi:10.1016/S0377-2217(03)00375-8.

Zhao, J., Yu, Y., Ren, H., Makowski, M., Granat, J., Nahorski, Z., Ma, T., 2022. How the power-to-liquid technology can contribute to reaching carbon neutrality of the China's transportation sector? Energy 261, 125058. Published online: August 22, 2022. DOI https://doi.org/10.1016/j.energy.2022.125058.