Group Decision and Negotiation
https://doi.org/10.1007/s10726-024-09915-8

Group Decision Making in Multiobjective Optimization: A Systematic Literature Review

Juuso Pajasmaa[1] · **Kaisa Miettinen**[1] · **Johanna Silvennoinen**[1]

Accepted: 25 November 2024

Abstract

Group decision making has been studied from several viewpoints and a variety of methods has been proposed. However, in the literature on solving multiobjective optimization problems, the main focus has been on supporting a single decision maker. We conducted a systematic literature review to examine and synthesize the state-of-the-art of multiobjective optimization methods developed for group decision making. We analyze group decision making methods of multiobjective optimization according to how preferences of several decision makers are incorporated into the solution process, how to select the most preferred solution for the group, different types of decision makers, types of groups and how the group is to operate during the solution process. In addition, we identify the key issues in the literature that are required to be considered in further method development to increase the methods' applicability in solving real-world problems. Finally, we guide how to select a method for solving real-world multiobjective optimization problems with multiple decision makers and suggest future research directions.

Keywords Multiple objective optimization · Preference information · Survey · Several decision makers · Consensus

1 Introduction

Multiobjective optimization problems (MOPs) contain several conflicting objective functions to be optimized at the same time. Objective functions depend on decision variables and solutions need to be generated before they can be evaluated. These problems do not typically have an optimal solution in which all the objective functions can reach their individual optimal values. Instead, MOPs have a set of so-called Pareto optimal solutions, meaning that there are trade-offs among the

✉ Juuso Pajasmaa
juuso.p.pajasmaa@jyu.fi

1 University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland

Published online: 18 December 2024

objective functions. Typically, solving a MOP means distinguishing a single Pareto optimal solution to be implemented in practice. Usually, this is achieved by acquiring some additional information from a domain expert, a decision maker (DM), who has insight into the problem and can articulate preference information related to the objective functions and solutions considered. The preference information is utilized in multiobjective optimization (MOO) methods to find the most preferred solution for the DM (Miettinen 1999; Miettinen et al. 2008, 2016).

However, many real-world problems involve more than one DM. Thus, if DMs solve the problem separately according to their individual preferences, they most likely end up with different solutions. Group decision making (GDM) involves multiple DMs with different preferences participating in solving a shared problem in which they must make a decision which is acceptable to all DMs (Kilgour and Eden 2010; Hwang and Lin 1987; Lu and Ruan 2007; Raiffa et al. 2002; Jelassi et al. 1990). The group shares responsibility for the final decision and may e.g., consist of individuals of a common collective (e.g., family, company, government) (Jelassi et al. 1990; Hwang and Lin 1987). The focus in this paper is in finding an acceptable solution to all DMs, and therefore we do not consider negotiation problems, where multiple parties may make a collective choice, or may “return to status-quo” and walk away without making any choice at all (Raiffa et al. 2002; Kilgour and Eden 2010).

The benefits of GDM to a decision making process include adding more resources and cognitive potential for making effective decisions (Steiner 1972). As drawbacks, participating in a GDM process may overload the DMs both in communication and increasing cognitive load, in addition to possible interpersonal conflicts (Raiffa et al. 2002; Steiner 1972). Furthermore, in collaborative groups, the desire for agreement among the group members may result in an irrational decision (Harvey 1974). To combat these issues, there is often a third party involved, called a moderator or a facilitator, helping to coordinate the GDM process (Steiner 1972; Raiffa et al. 2002).

When considering decisions with multiple conflicting objective functions or criteria involving either a single DM or multiple DMs, there are different types of problems: MOO type of problems and multicriteria decision analysis (MCDA) problems.[1] In MCDA problem, the set of solution alternatives is explicitly known, discrete and finite (Hwang and Masud 1979; Miettinen 1999). Regarding GDM, MCDA type of problems are commonly known as multi-criteria GDM (MCGDM) (Belton and Stewart 2003; Rezaei 2015). Generally, MCGDM methods focus on aggregating the preferences of the DMs (of the alternative solutions) in some way to a collective preference, which is then exploited to select the best one (Boix-Cots et al. 2023; Hwang and Lin 1987; Keeney et al. 1993) (c.f. (Boix-Cots et al. 2023) for more details on MCGDM and preference aggregation). There is a lot of literature on MCGDM (Hwang and Yoon 1992; Chen and Hwang 1992), but not in MOO type of problems, as noted also in the bibliometric analyses on GDM literature (Laengle et al. 2018; Wang et al. 2021).

[1] Although this review focuses on MOO, we mention MCDA type of problems as they offer relevant background and are also utilized in the reviewed literature.

Despite the real-world relevance and practical need to involve the preferences of multiple DMs in solving MOPs, appropriate methods have not received much attention in the literature. To the authors' best knowledge, no systematic literature review focusing on methods for these problems exists. Therefore, we provide an overview of *multiobjective optimization* methods developed for GDM. We refer to group decision making in multiobjective optimization as GDM-MOO, for short. The term referring to the corresponding problem, GDM-MOP, was first used by Xiong et al. (2013) and thereafter also in emerging literature (Balderas et al. 2022; Fernández et al. 2022, 2021; Nag et al. 2018; Sakamoto et al. 2021, 2022).

When developing GDM-MOO methods, there are new challenges to be addressed when compared to methods for a single DM: How to support DMs in handling the complexity of balancing multiple conflicting objective functions? How to deal with the conflicting preferences of the DMs? How to support them in getting closer in their preferences to be able to find a solution that they all can accept via e.g., GDM processes to support collaboration among the DMs in group discussions? How to support different group structures? Or how to eventually select the final solution? And how to demonstrate the applicability in real-world problems? Furthermore, the methods should not add an excessively high cognitive load on DMs or interfere with the solution process, e.g., by biasing DMs or directing the search in a certain direction.

This systematic literature review examines the state-of-the-art of GDM-MOO methods and their main working principles to clarify how these methods are used to solve GDM-MOPs. We applied the PRISMA procedure (Page et al. 2021) and eventually 31 papers were included. This paper provides novel classifications and presents relevant trends in the current literature, as well as sheds light on dismissed elements of solution processes, that cannot be ignored when the applicability of the methods in practice is of concern. We classify GDM-MOO methods, different types of groups of DMs, ways to aggregate preferences, and describe solution processes in different types of GDM-MOO methods. The classifications help the readers (both researchers and practitioners) to get a grasp of the different approaches available for solving GDM-MOPs. Furthermore, for researchers, the knowledge of the key issues in the current literature on GDM-MOO methods help further research to be devoted to address them.

The main questions to be answered can be explicitly stated as:

- What kind of MOPs have been solved in GDM?
- What kind of methods have been developed and how are the multiple preferences from several DMs incorporated into the optimization process?
- What methods are used to support the group to select the final solution?

This systematic literature review results in classifying GDM-MOO methods in three main classes: a priori, a posteriori and interactive methods based on the timing of providing preference information. The classification corresponds to the one used for cases with a single DM (Hwang and Masud 1979; Miettinen 1999).

The rest of the paper is structured as follows. In Sect. 2, we summarize the key concepts and terminology used in the paper, including the proposed classifications related to GDM-MOO. We describe the procedure of the systematic literature review in Sect. 3 and provide an overview of the results in Sect. 4. In Sects. 5 and 6, we present the main ideas of the methods classified to non-interactive and interactive methods, respectively. In Sect. 7, we discuss key takeaways from the reviewed literature as well as key issues to tackle in developing GDM-MOO methods. Finally, we conclude in Sect. 8 and raise some future research directions.

2 Key Concepts and Terminology

This section covers the key concepts and terminology for later use in this paper. We define a MOO problem and ways for DMs to articulate preference information and preference modeling in Subsection 2.1. Then, in Subsection 2.2, we introduce the key concepts of GDM-MOPs and GDM-MOO methods and the relevant aspects that are used to classify the papers.

2.1 Key Concepts of Multiobjective Optimization

A multiobjective optimization problem can be formulated as follows:

$$\begin{aligned} &\text{minimize } \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ &\text{subject to } \mathbf{x} \in S, \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, \dots, x_n)^T$ is called a *decision variable vector*, and it belongs to S known as a *feasible set*. The feasible set is constructed with constraints in the *decision space* $\mathbf{R}^n$. A (feasible) decision variable vector is also called here a solution. The vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ represents objective function values constructing an *objective vector* in the *objective space* $\mathbf{R}^k$.

A solution is Pareto optimal if no values of any objective functions can be improved without declining some other objective function values, meaning that there are trade-offs among the objective functions. An objective vector $f(x_1)$ is said to dominate another objective vector $f(x_2)$, if the objective vector $f(x_1)$ is strictly less in at least one objective value and less or equal in others (Steuer 1986). Sometimes, solutions that do not dominate each other are referred to as non-dominated solutions. Furthermore, an ideal vector in the objective space contains the best feasible values of each objective function in the set of Pareto optimal objective vectors. The set of Pareto optimal objective vectors is sometimes called a Pareto front.

Finding a solution to a MOP in one way or another is called a solution process (Miettinen 1999). In this review, solving a MOP means selecting the most preferred Pareto optimal solution by a DM as the final solution to be implemented in practice. Different types of MOO methods exist, and they can be classified according to the role of the DM in the solution process (Hwang and Masud 1979; Miettinen 1999) in no preference methods (where no preference information is available), a priori methods (where a DM articulates their preferences before optimization), a posteriori methods (where a representative set of Pareto optimal solutions is generated for a

DM to select from) and interactive methods (where a DM iteratively articulates their preferences during the solution process).

In what follows, we describe the classes briefly and give some examples of methods concentrating on those that will be referred to later in this paper. Without any preference information, the final solution has to be selected by using some global criterion or select some neutral compromise solution (Miettinen 1999). In a priori methods, the DM must have a good understanding in advance about the problem to be able to specify accurate preference information. A posteriori methods require the DM to be able to select a solution from a large set of solutions, which can be cognitively demanding.

Interactive methods consist of iterative steps repeating phases of preference articulation and solution generation alternating until the DM has found the most preferred solution (or some other stopping criterion is met or there are no more solutions to be found). This allows the DM to learn about the trade-offs, what kind of solutions are available and how feasible their preferences are. However, the solution process may be time-consuming for the DM and these methods should be applied only when a DM is able to play a role in the solution process (Afsar et al. 2021; Belton et al. 2008; Miettinen et al. 2008, 2016). Interactive methods involve either a technical stopping criterion (such as the number of iterations or the number of function evaluations), or a stopping criterion based on the satisfaction of the DM, where the DM can freely stop when the most preferred solution has been found.

Among MOO methods used to solve MOPs, we have scalarization-based and evolutionary methods (Miettinen 2008; Deb 2001, 2008). In scalarization-based methods, the MOP is converted to a single objective (scalar-valued) optimization problem (Miettinen 1999). The new scalarized problem with a real-valued objective function can be solved with an appropriate single-objective optimization method. When properly done, scalarization can guarantee the Pareto optimality of the found solution to the original MOP (Miettinen 1999; Miettinen et al. 2016; Sawaragi et al. 1985). It has to be kept in mind that the solution found is significantly impacted by the choice of the scalarization function (Miettinen and Mäkelä 2002).

There are various scalarization-based methods in the literature, such as goal programming (Charnes and Cooper 1961, 1977) and ϵ-constraint-method (Haimes et al. 1971). Goal programming methods minimize the sum of deviations from the DM's preferences to find a solution, while in the ϵ-constraint-method, one of the objective functions is selected to be optimized, while the others are converted into constraints.

Population based evolutionary methods were originally of a posteriori type. The population of candidate solutions is manipulated in iterations called generations so that the next generation would approach Pareto optimality and cover the Pareto front diversely (Bäck 1996; Deb 2008). Evolutionary methods can also be of a priori type. Then, rather than covering the entire Pareto front, they focus on its subset considering the preferences of the DM. Similarly, interactive evolutionary methods can be seen as iterative steps of preference articulation and evolutionary solution generation. Evolutionary methods are less likely to get stuck on local optima (Coello et al. 2007; Deb 2001; Deb et al. 2002) than some scalarization-based methods. However, as they are metaheuristics, (Bäck 1996; Deb 2001), they can only guarantee the nondominance of the solutions in the final population. In addition, evolutionary methods have several parameters (such as population size, evolutionary operators to use, their inner parameters and their probabilities) that need to be specified and affect the performance (Jin 2005).

One of the widely used a posteriori evolutionary methods is NSGA-II (Deb et al. 2002). It selects the population members that survive to the next population by sorting solutions to ranks where solutions in the same rank are mutually nondominated and also considering the coverage of the solutions. Many NSGA-II-based GDM-MOO methods modify the way to select the population members to be applicable for problems with multiple DMs. Finally, we have to clarify that some of the interactive evolutionary GDM-MOO methods (e.g., (Kadziński and Tomczyk 2017)), run for a *predetermined amount of generations*, after which, the DMs articulate their preferences. For example, iteratively after every 100 generations, the DMs articulate their preferences, until a stopping criterion is met.

Preference articulation and modeling preferences

DMs may prefer to articulate their preference information in different ways. For simplicity, we describe preference information types from the perspective of a single DM, although this also applies to multiple DMs.

One preference information type is a so-called *reference point* (Wierzbicki 1982). It is in the objective space and contains an aspiration level that the DM would like to achieve for each objective function. A DM can also articulate their preferences via comparisons of selected solutions: *pairwise comparisons of solutions*, *classifying solutions*, selecting the *most preferred solution* and *ranking solutions*. The pairwise comparisons mean that the DM evaluates the relation of two solutions, either one solution is preferred to the other or the two solutions are indifferent (Branke et al. 2015). The classifying of solutions means that the DM classifies solutions in terms of how satisfactory they are to classes such as highly satisfying, satisfying, dissatisfying and highly dissatisfying. The most preferred solution selection means that the DM chooses the best solution (according to their preferences) from a set of solutions. In ranking, the DM ranks the solutions from the most to the least preferred (Coello 2000).

The DM can compare the objective functions, reflecting their relations. Examples of this are the *classification of objective functions*, allocating *weights for objective functions* and *ranking of objective functions*. The classification of objective functions means splitting the objective functions into several classes according to the types of desirable changes for the objective function values (Xin et al. 2018). Weights for objective functions are supposed to represent their relative degrees of importance for a DM. However, it is not in practice clear what this actually means as the weights behave in an indirect way e.g., if objective functions have correlations (Miettinen et al. 2008; Roy 1971; Steuer 1986). It has been shown to be very difficult for a DM to adjust the weights to obtain a desirable solution (Nakayama 1995; Steuer 1986; Wang et al. 2017). The ranking of objective functions means ranking them according to the DM's preferences. This can be achieved for example using natural language (Baril et al. 2012) or using MCDA methods.

As discussed earlier, MCDA problems have an explicitly known solution set, where each solution is characterized by multiple criteria or attributes and there are only a few solutions in total. MCDA methods construct a preference model, which is a quantitative description of a DM's preferences, from the DM's preference information (Xin et al. 2018; Keeney et al. 1993; Li and Hu 2023). The preference model is exploited in different methods to the alternative solutions either by choosing, ranking or sorting (Lu and Ruan 2007; Chen and Hwang 1992; Keeney et al. 1993; Öztürk et al. 2005).

According to Roy (1996); Jacquet-Lagreze and Siskos (2001), MCDA methods can be roughly split into value-focused methods and (out)ranking methods. The former apply a utility or value function to model the DM's global preferences and assume that the DM makes decisions based on an underlying utility or value function (Keeney et al. 1993) by maximizing it. For example, in multi-attribute utility theory (Keeney et al. 1993), the DM's preferences are modeled with a utility function, according to which the different alternative solutions can be rank-ordered. The utility function can be estimated from e.g., pairwise comparisons of solutions. In analytic hierarchy process (AHP), pairwise comparisons are used to build relative importance relations among the attributes and alternatives (Saaty 1980).

Outranking methods construct binary relations, referred to as outranking relations, among the alternatives to represent the DM's preferences. These methods aim to determine for each pair of solutions whether the DM prefers one of them, is indifferent or the solutions are incomparable. To achieve this, different preference indicators are defined and compared with specific threshold values, such as preference and veto thresholds for each criterion (which the DM must give as preference information) (Figueira et al. 2005). An example of outranking methods is ELECTRE (Roy 1971), in which outranking relations are exploited to recommend the best alternative, the best order or ranking of the alternatives depending on the problem in question.

2.2 Key Concepts of GDM-MOO

In what follows, we discuss the key concepts of GDM-MOO relevant to this review. We propose a classification to identify different types of groups that can play a role in GDM-MOO and a classification of ways to aggregate preferences for GDM-MOO. Finally, we describe the main approaches used in solving GDM-MOPs in the reviewed literature.

This paper considers a GDM-MOP as a combination of a GDM and MOP. The multiple DMs involved in the solution process are solving a common MOP, meaning that the DMs share the same decision variables, objective functions and constraints. However, the individual DMs may have conflicting preferences on the objective functions and different ideas on what is the most preferred solution for the group. As mentioned, GDM-MOO methods can be classified to a priori, a posteriori and interactive methods (Nag et al. 2018), similarly to MOO methods for a single DM (Hwang and Masud 1979; Miettinen 1999).

GDM-MOO methods incorporate and aggregate the DMs' preferences in some manner to find solutions considering the given preferences. In the literature, these solutions are referred to in various terms, such as consensus solutions (Pfeiffer et al. 2008; Sakamoto et al. 2022; Fernández et al. 2021), compromise solutions

(Hadas and Nahum 2016) and collective solutions (Borissova and Mustakerov 2017; Li and Hu 2023; Fernández et al. 2022). We propose using the term *collective solutions* as it is neutral and does not imply that all DMs should find them acceptable or that some kind of a consensus measurement has been conducted among the DMs regarding the solutions. However, collective solutions are not necessarily always Pareto optimal (although often they are) as some methods relax the optimality requirement to match the groups' preferences better.

By solving a GDM-MOP, we mean distinguishing a final solution among collective solutions so that the group accepts it. The underlying assumption is that the DMs at least to some degree collaborate during the solution process due to the responsibility of finding the final solution.

Structuring the group of DMs

Individual DMs may form different kinds of groups that may communicate in various ways (in-person or online) and share preference information with some DMs or with the whole group. Additionally, there can be different roles among the DMs of the group. We refer to these aspects as a *group structure*. Roughly speaking, two main group structures emerge in the literature (Hwang and Lin 1987; Jelassi et al. 1990; Raiffa et al. 2002; Marakas 2003): an advisory group for a unitary DM (referred to as a supra DM (SDM) according to Keeney et al. (1993)) and an established group.

In an advisory group, the group is advising the SDM who is in charge of selecting the final solution. The group members are not directly responsible for the final solution and its effects in reality (Hwang and Lin 1987; Jelassi et al. 1990; Raiffa et al. 2002). The advisory group can be further divided into *teams* and *committees*, based on the role of the SDM in the GDM process (Marakas 2003; Raiffa et al. 2002). In a team, a SDM is leading the GDM process by actively participating in the solution process. The group members only communicate and share their preference information with the SDM. An example of a team is a company's team of experts led by the boss choosing the best investment policy. On the other hand, in a committee, the DMs communicate with each other during the solution process, possibly sharing also preference information. After selecting the final solution, they propose it to the SDM who either accepts it or requires the group to continue the solution process. An example of a committee is a governmental body drafting a suggestion for a legislation change.

In an *established group*, each DM is in part responsible for the final solution and its effects in reality (Hwang and Lin 1987; Jelassi et al. 1990; Raiffa et al. 2002). Since there is no SDM in charge of the solution process, in some way (implicitly or explicitly) before the solution process starts, the group has to decide the rules of the solution process, involving communication, sharing of preference information and the method for selecting the final solution. For example, the group may prefer to discuss freely, but not share their preference information and select the final solution from a set of collective solutions via a majority vote. An example of an established group is a set of a company's directors making a plan for the company's future.

Figure 1 visualizes the main aspects of different group structures. The circles are DMs, the star refers to a SDM and the lines depict the flow of information. The flow

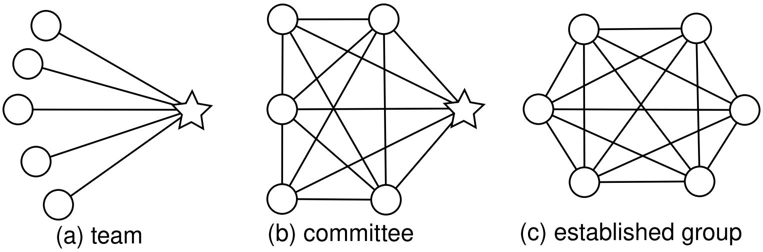


Fig. 1 Group structures introduced in Marakas (2003): (**a**) team, (**b**) committee and (**c**) established group

of information includes the communication among the DMs and in which way the preference information is shared.

All these group structures allow giving different degrees of importance to the DMs, often modeled with weights. For brevity, we refer to the existence of different degrees of importance of DMs, as *importance of DMs*.

Classification of the ways to aggregate preferences

Based on the findings of the reviewed literature and existing GDM classifications, we propose the following classification of different ways to aggregate preferences in GDM-MOO. The classification is not strict but describes different approaches in the literature and is inspired by Boix-Cots et al. (2023). In what follows, we refer to methods of types A–D:

- A: Methods focusing on *consensus reaching processes* (CRP) to reach a consensus.
- B: Methods aggregating the preferences into a collective preference and then, finding solutions according to that collective preference.
- C: Methods incorporating the preferences into the search and then, finding solutions according to more than one preference.
- D: Combined methods

Type A methods rely on the DMs, willingly or unwillingly, to adjust their preferences during the solution process to an agreement or a consensus. This class includes group discussions and CRPs that consist of several rounds of discussions among the DMs designed to reach a consensus before making a decision (Saint and Lawson 1994; Labella et al. 2018). Reaching a consensus implies that the DMs should adjust their preferences to be more similar to the preferences of the group (Labella et al. 2018). A third party (e.g., a moderator, an SDM or a decision support system) coordinates the process and helps the DMs adjust their preferences by giving some feedback or by adjusting in an automated fashion the DMs' preferences to be more similar (Cabrerizo et al. 2010; Herrera-Viedma et al. 2014; Labella et al. 2018; Palomares et al. 2014; Chao et al. 2022; Herrera et al. 1995, 1996). These automated feedback methods often use various preference models representing the DMs individually and as a group and measure the level of agreement of the DMs on a given solution by a *consensus measure*. Most consensus measures are based on distances to the collective preference or based on distances between the preferences of the DMs (Palomares et al. 2014). Consensus measures can also be used without an explicit CRP involving interaction among the DMs.

Type B methods focus on aggregating the preference information articulated by the DMs into a collective preference. Then, the collective preference directs the search in the optimization and is exploited to find solution(s) from which the final solution is selected. The aggregation is done with aggregation operators such as mean, and they can include consensus measures, fuzzy sets, linguistic information or incorporate the DMs' importance e.g., by using a weighted mean of the preferences (Boix-Cots et al. 2023).

Type C methods focus on handling preferences from different DMs, which we call multiple preferences. The multiple preferences direct the search in the optimization and in this way, multiple solutions are found. Type C methods can also incorporate a collective preference. Often, the preferences are or are converted to rankings, orderings or sortings of solutions to determine the best solution(s). These methods include also voting methods and MCDA methods. Finally, type D methods refer to methods combining two or more of the previously presented types.

Main approaches of solving GDM-MOPs

The reviewed literature proposes different GDM-MOO methods and here we describe the main approaches on a general level. Overviews of a priori, a posteriori and interactive GDM-MOO methods are visualized in Figs. 2, 3 and 4, respectively. In what follows, we introduce the symbols used. A big double pointed arrow refers to information flowing to a type A method and information flowing back to

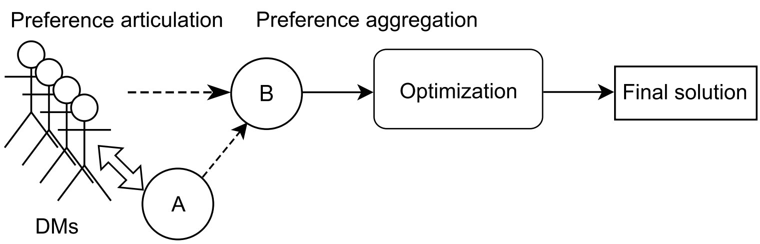


Fig. 2 Flowchart of a general solution process in a priori GDM-MOO methods

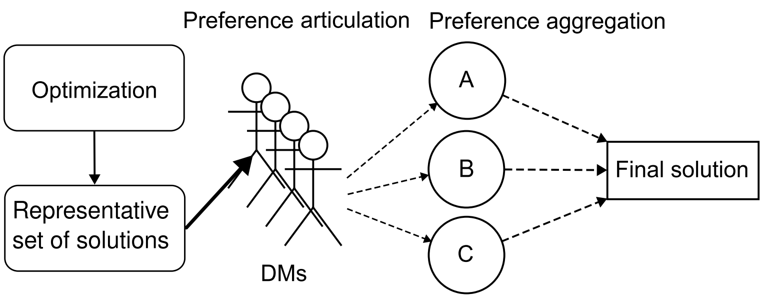


Fig. 3 Flowchart of a general solution process in a posteriori GDM-MOO methods

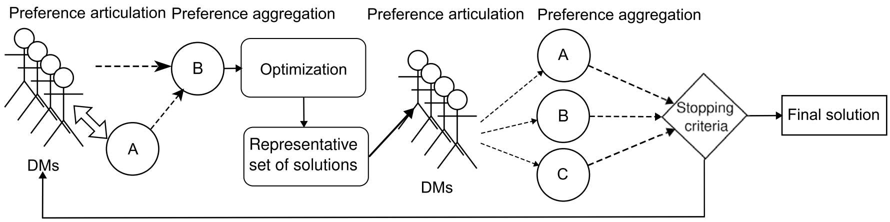


Fig. 4 Flowchart of a general solution process for an interactive GDM-MOO methods

the DMs. Dashed lines with arrows refer to the option of using different preference information types or ways to aggregate preferences. A solid arrow refers to mandatory actions.

In a priori GDM-MOO methods (see Fig. 2), before the DMs articulate their preferences, a type A method can be applied to make DMs preferences more similar. Then, the DMs articulate their preferences and type B methods aggregate the preferences to a collective preference. The collective preference is then exploited in the optimization to find a solution to be selected as the final solution.

In a posteriori GDM-MOO methods, first a representative set of solutions is found by an optimization process, and then it is shown to the DMs (see Fig. 3). If this set is large, some filtering has to be done if the DMs' are expected, for example, to rank all the solutions. Otherwise, the whole set can be shown. Then, the DMs articulate their preferences and by applying either a type A, B or C method the final solution is selected.

In interactive GDM-MOO methods (see Fig. 4), we should note that some of the methods do not require the DMs to articulate their preferences in each iteration, see e.g., (Kadziński and Tomczyk 2017; Tomczyk and Kadziński 2022). In addition, the methods use the following stopping criteria: (a) a stopping criterion based on the satisfaction of the DMs and (b) a technical stopping criterion. Lastly, there are two main approaches for selecting the final solution: approaches where the final solution selection is an *integral part of the method* and approaches where the final solution selection is done in a *post-processing* stage after the actual solution process.

We can identify nine steps in interactive GDM-MOO methods divided to stages before and after optimization. The steps determine, e.g., how and when the DMs' preferences are incorporated in the solution process.

Before optimization

- Step 1: Show some problem information to the DMs. Optionally, apply a type A method aiming to make the preferences of the DMs more alike.
- Step 2: Ask the DMs to articulate their preferences.
- Step 3: If a type B method is applied, aggregate the DMs' preferences to a collective preference. Otherwise, incorporate the multiple preferences of the several DMs into the optimization process. This includes building preference models for each DM, if applicable.
- Step 4: Conduct optimization with the information obtained.

After optimization

- Step 5: Show a representative set of solutions to the DMs.
- Step 6: Ask the DMs to articulate their preferences and aggregate them via a type A, B or C method.
- Step 7: Stopping criterion
  - If stopping criterion (a) is met:
    - *If finding the final solution is integrated in the method*, the best solution according to the method is the final solution, and go to step 9.
    - *If post-processing is needed*, conduct an additional phase to select the final solution and go to step 9. If the post-processing ends unsuccessfully, go to step 1 for another type A method or to step 2.
  - If stopping criterion (b) is met:
    - *If finding the final solution is integrated in the method*, the best solution according to the method is the final solution, and go to step 9.
    - *If post-processing is needed*, conduct an additional phase to select the final solution and go to step 9.
- Step 8: Otherwise, go to step 1 or step 2.
- Step 9: The solution process ends.

3 Systematic Literature Review

This section describes how the systematic literature review was performed. The review followed the PRISMA procedure (Page et al. 2021) of i) identifying relevant records from the selected databases, ii) screening and iii) inclusion. We start by describing the rationale for the review including the aims of the review and the research questions. Next, we specify the inclusion and exclusion criteria used in the review, all the databases and keywords used and then describe the search and selection process. Finally, we provide justifications for the different classifications used in this review according to which the findings are presented in the following section.

The search process: The databases used in the literature searches were Scopus and Web of Science. After selecting the databases, the search queries were formed by performing a few test searches using test queries. Then, with the search queries listed in Table 1, the searches for the review were performed. The searches contain papers published until end of 2023. Differences in the search queries between Web of Science and Scopus are related to the syntax of the search engine, but they perform functionally in the same way.

Inclusion criteria: A paper was included if it did contain relevant combinations of the keywords in the title, or abstract. In addition, the paper had to contain human DMs involved in GDM and solving a MOP. The DMs' involvement means that the DMs' preferences were incorporated into the optimization in an a priori,

Table 1 The search queries for Scopus and Web of Science

Scopus	Web of Science
(TITLE-ABS-KEY ("group decision" OR "group decision making" OR "group preference")) AND TITLE-ABS-KEY ("multiobjective optimization" OR "multi-objective optimization" OR "multiple objective optimization" OR "multi-objective decision making") AND NOT ALL ("multi-attribute")	TS=("group decision" OR "group decision making" OR "group preference") AND TS=("multiobjective optimization" OR "multi-objective optimization" OR "multiple objective optimization" OR "multi-objective decision making") NOT ALL=("multi-attribute")

a posteriori, interactive or in some other manner and the DMs needed to play an active role in the solution process.

Exclusion criteria: A paper was excluded if it was not written in English or did not fulfill the inclusion criteria. In practice, this means that papers not focusing on multiobjective *optimization* problems were excluded such as multi-attribute or MCGDM problems containing a predetermined and explicitly given set of alternatives. Further exclusions were made when the terminology used in the paper was not defined clearly or did not follow the conventions of the field of MOO.

Figure 5 contains a PRISMA flowchart showing graphically the steps described next. The search queries resulted with 128 papers from Scopus and 118 papers from Web of Science. Before screening, 38 duplicate records were removed and six other records were removed (not published papers). In total, before screening, there were 204 papers left.

The screening was conducted in two phases: i) pre-screening, where the title, keywords and abstract were reviewed manually to determine whether some exclusion criterion was met, ii) screening with another round of careful evaluation of the title, keywords and abstract including the inclusion criteria. The PRISMA procedure includes one screening phase but for our review, a two-phase screening procedure was conducted due to the manual reviewing process combined with the amount of papers to be examined for the screening phase (to be less error-prone). After the pre-screening and screening phases, 67 papers were sought for retrieval from which 64 papers could be retrieved.

Next, the 64 papers were assessed for inclusion with full-text reviews again according to the two-phased procedure. In the first phase, a briefer full-text review was performed to see if the paper superficially fulfilled the inclusion criteria. In this phase, 28 papers were excluded (reason 1). In this phase, also the forward and backward searches of the included papers were performed. This brought up 8 new papers to be included in the review, resulting with a total of 44 papers.

In the second phase, a more detailed examination of the included papers was performed considering the research questions of the literature review. During this process, it was noticed that several papers lacked much of relevant information, for example, it turned out that the alternatives were known and not based on optimization. This led to nine more papers being excluded (reason 2). Additionally, four more papers were excluded because the terms were not defined clearly and

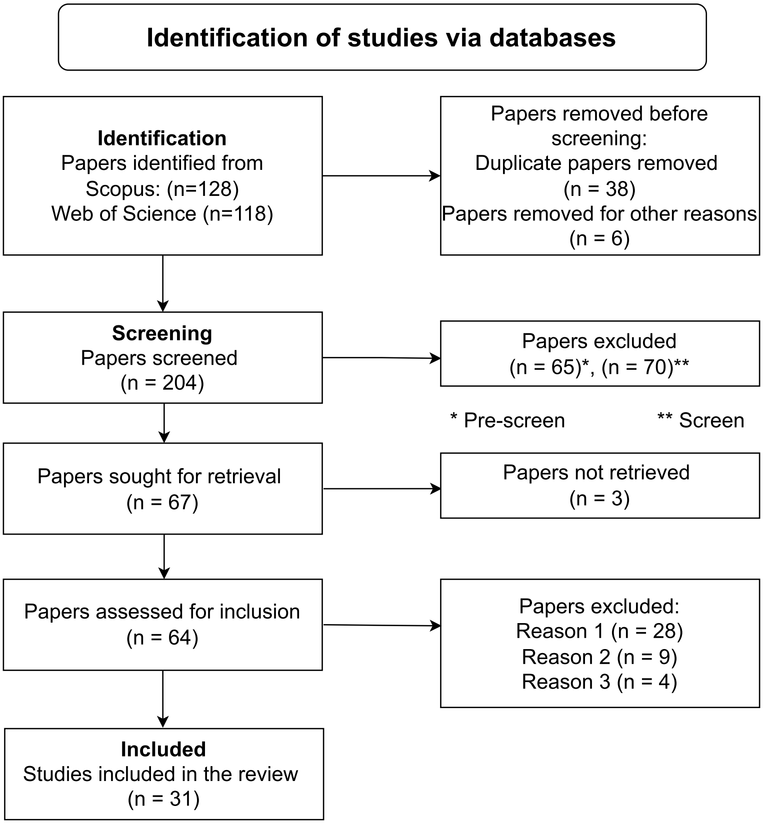


Fig. 5 A flowchart showing the PRISMA steps taken when conducting the literature review (adapted from Page et al. (2021))

did not follow the conventions of the field (reason 3). Eventually, a total of 31 papers constituted the body of literature to be reviewed very carefully.

4 Overview of the Literature

As mentioned, this systematic literature review consists of 31 papers with the focus on GDM-MOO methods for different types of GDM-MOPs. One should note that some papers propose multiple variants of the developed methods considering different types of groups or types of DMs or type of preference information or they solve different types of problems. Hence, the sums of the numbers of methods in the subsequent figures do not always match the mentioned number of papers. Next, we define the main classes according to which the results are presented.

The types of problems to be solved need to be defined. We divide MOPs considered in the papers into *test problems*, *real-world inspired problems* and *real-world* *applications*. By test problems, we mean various bench-marking problems, where the objective functions and decision variables do not typically have any reasonable names or meanings. Additionally, the exact shapes and locations of Pareto fronts are known. On the other hand, in real-world inspired problems, objective functions and decision variables have meanings and the formulations are realistic in the sense of the related domain, but the exact setting may not exist or be fully real. Finally, real-world applications refer to real-world problems, where the authors have access to the relevant data to form a real problem setting.

In the considered papers (see Fig. 6), the proposed methods are tested with test problems in 14 papers (Bechikh et al. 2011; Pfeiffer et al. 2008; Xin et al. 2018; Nag et al. 2014, 2018; Liu et al. 2011; Chiu et al. 2019; Cinalli et al. 2015; Tomczyk and Kadziński 2022; Kadziński and Tomczyk 2017; Wendell 1980; Oliveira and Ferreira 2000; Sakamoto et al. 2021, 2022) and with real-world inspired problems also in 15 papers (Ahmad et al. 2022; Bechikh et al. 2013; Subulan et al. 2015; Pfeiffer et al. 2008; Chiu et al. 2019; Lewis and Butler 1993; Cinalli et al. 2015, 2020; Wu et al. 2007; Baril et al. 2012; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Tomczyk and Kadziński 2022) with some papers solving both types of problems. Six papers out of 31 papers solve real-world applications, see (Baharmand et al. 2020; Li and Hu 2023; Varas et al. 2020; Borissova and Mustakerov 2017; Maharjan and Hanaoka 2018; Hadas and Nahum 2016). Regarding the number of objective functions in the solved problems, a majority of the papers (21 in total) (Maharjan and Hanaoka 2018; Ahmad et al. 2022; Bechikh et al. 2013; Pfeiffer et al. 2008; Xiong et al. 2013; Nag et al. 2014, 2018; Liu et al. 2011; Borissova and Mustakerov 2017; Chiu et al. 2019; Varas et al. 2020; Li and Hu 2023; Cinalli et al. 2015, 2020; Baril et al. 2012; Kadziński and Tomczyk 2017; Tomczyk and Kadziński 2022; Wendell 1980; Oliveira and Ferreira 2000; Sakamoto et al. 2021, 2022) solve biobjective problems. Problems with three objective functions are solved in nine papers (Subulan et al. 2015; Baharmand et al. 2020; Hadas

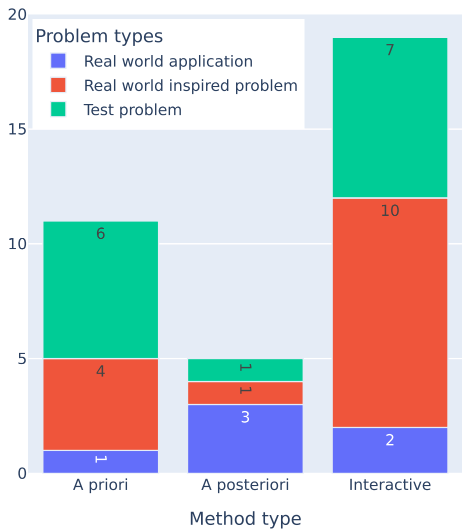


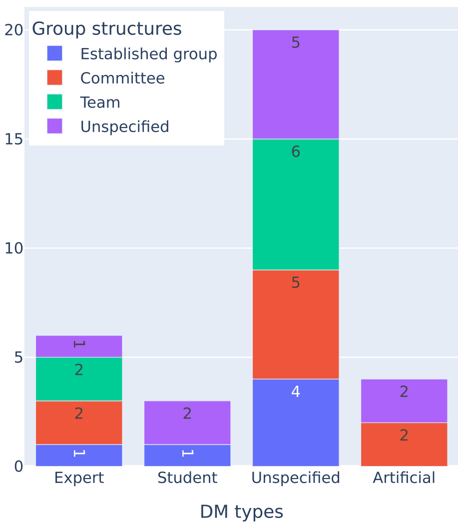
Fig. 6 Problem types in the reviewed papers

and Nahum 2016; Chiu et al. 2019; Lewis and Butler 1993; Cinalli et al. 2015; Wu et al. 2007; Kadziński and Tomczyk 2017; Tomczyk and Kadziński 2022). Only six papers (Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Kadziński and Tomczyk 2017; Tomczyk and Kadziński 2022) solve problems with more than 4 objective functions. To summarize, the majority of the literature is very focused on problems with 2–3 objective functions and solving test problems or real-world inspired problems instead of real world applications.

We classify different types of DMs to *artificial*, *student*, *expert* and *unspecified* DMs. By artificial DMs, we mean DMs that are not human beings, for example, agents or application of decision rules such as a utility function to provide DMs' preferences. Using artificial DMs naturally implies that the human interaction parts of GDM are not considered. As the name suggests, student DMs refer to students acting as DMs and expert DMs are real domain experts solving their problem. The latter means that the DMs are really responsible for the final solution and the effects it has in reality. Finally, unspecified DMs refer to cases where the authors do not specify who the DMs are but some specific properties or behavior of them are still assumed. In most (i.e., 20) papers, the DMs are unspecified, see Fig. 7. Expert DMs are used in seven papers (Li and Hu 2023; Borissova and Mustakerov 2017; Maharjan and Hanaoka 2018; Baharmand et al. 2020; Hadas and Nahum 2016; Varas et al. 2020; Subulan et al. 2015), artificial DMs in four papers (Sakamoto et al. 2021, 2022; Wendell 1980; Oliveira and Ferreira 2000) and students in three papers (Cinalli et al. 2015, 2020; Lewis and Butler 1993). In addition, in six papers (Bechikh et al. 2013; Subulan et al. 2015; Chiu et al. 2019; Borissova and Mustakerov 2017; Li and Hu 2023; Wu et al. 2007) different importance of the DMs are considered, modeled with weights.

As far as group structure is concerned, in 11 papers it is not specified, see Fig. 7. Nine papers consider committees (Li and Hu 2023; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Kadziński and Tomczyk 2017; Tomczyk and

Fig. 7 Group structures and DM types in the reviewed papers



Kadziński 2022; Wendell 1980; Oliveira and Ferreira 2000; Maharjan and Hanaoka 2018), eight papers consider teams (Wu et al. 2007; Baril et al. 2012; Fernández and Olmedo 2013; Fernández et al. 2022; Borissova and Mustakerov 2017; Chiu et al. 2019; Ahmad et al. 2022; Subulan et al. 2015) and six papers consider established groups (Lewis and Butler 1993; Fernández and Olmedo 2013; Balderas et al. 2022; Baharmand et al. 2020; Bechikh et al. 2011, 2013). It is worth mentioning that the classification to group structures is not typically provided directly in the papers. Therefore, this information is mainly derived from clues or due to the authors using these specific terms loosely. However, due to the large amount of the papers having either unspecified DMs, group structures or both, much of the behavior of the DMs or the group may have been ignored in the current literature.

Briefly, on the GDM-MOO method types, proposing interactive methods is most popular (17 papers) (Varas et al. 2020; Li and Hu 2023; Lewis and Butler 1993; Cinalli et al. 2015, 2020; Wu et al. 2007; Baril et al. 2012; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Tomczyk and Kadziński 2022; Kadziński and Tomczyk 2017; Wendell 1980; Oliveira and Ferreira 2000; Sakamoto et al. 2021, 2022) and then a priori methods (10 papers) (Maharjan and Hanaoka 2018; Subulan et al. 2015; Ahmad et al. 2022; Bechikh et al. 2011, 2013; Pfeiffer et al. 2008; Xiong et al. 2013; Nag et al. 2014, 2018; Liu et al. 2011) and finally a posteriori methods (4 papers) (Baharmand et al. 2020; Borissova and Mustakerov 2017; Hadas and Nahum 2016; Chiu et al. 2019). It is not surprising that a majority of the literature focuses on interactive or a priori methods since GDM is by nature an interactive process and applying ways to aggregate preferences prior to optimization is a straightforward approach to incorporate multiple DMs in solving GDM-MOPs.

For a summary of the preference information types used in the papers, see Fig. 8. The two most commonly used preference information types are weights for objective functions in 10 papers (Sakamoto et al. 2021, 2022; Borissova and Mustakerov 2017; Hadas and Nahum 2016; Liu et al. 2011; Ahmad et al. 2022; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022) and reference

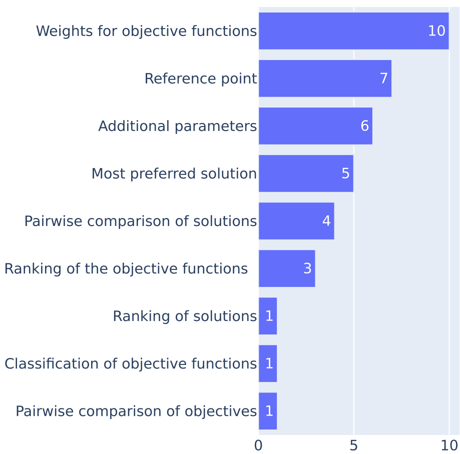


Fig. 8 The preference information types in the reviewed papers

points in seven papers (Li and Hu 2023; Nag et al. 2014, 2018; Xiong et al. 2013; Pfeiffer et al. 2008; Bechikh et al. 2011, 2013). In six papers (Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Li and Hu 2023; Pfeiffer et al. 2008), the DMs had to provide some additional parameters. The most preferred solutions are used in five papers (Wu et al. 2007; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022), however, in (Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022) the most preferred solution is converted to a reference point when utilized in the proposed GDM-MOO method. Pairwise comparisons of solutions are used in four papers (Kadziński and Tomczyk 2017; Tomczyk and Kadziński 2022; Cinalli et al. 2015, 2020), while ranking of objective functions is used by Chiu et al. (2019); Borissova and Mustakerov (2017), classification of objective functions by Baril et al. (2012), ranking of solutions by Lewis and Butler (1993), and pairwise comparisons of objective functions by Hadas and Nahum (2016). Many methods utilize preference models to perform (some or majority of) preference articulations instead of the DMs (Li and Hu 2023; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Kadziński and Tomczyk 2017; Tomczyk and Kadziński 2022; Cinalli et al. 2020, 2015). Moreover, some methods require DMs to articulate complex preference information, like fuzzy membership functions (Nag et al. 2018) or additional parameters for interval outranking models (Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022) or provide values to method's inner parameters (Pfeiffer et al. 2008; Li and Hu 2023), which affect the results a lot. (And typically DMs cannot be assumed to be able to do this since they are domain experts, not experts of the methods.) Finally, some methods contain very specific assumptions such as that DMs' preferences as a group follow a Gaussian distribution (Cinalli et al. 2015, 2020). While a majority of the papers use well-known preference information types, how they are used in the methods often seem very unclear for the DMs and the amount of expertise required from the DMs' regarding the method may be too much.

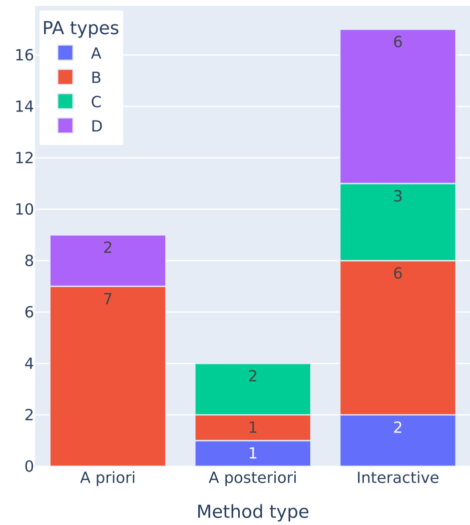
Figure 9 visualizes the MOO method types divided roughly into scalarization-based and evolutionary methods. A variety of scalarization methods are used in the current literature. In three papers a weighted sum (Maharjan and Hanaoka 2018; Ahmad et al. 2022; Borissova and Mustakerov 2017) is used and in three papers goal programming (Subulan et al. 2015; Wu et al. 2007; Baril et al. 2012) is utilized. In two papers different ε-constraint methods (Baharmand et al. 2020; Varas et al. 2020) and in two papers Tchebyscheff scalarization function (Varas et al. 2020; Lewis and Butler 1993) are used. Other scalarization-based methods include the WIN-method (Chiu et al. 2019), lexicographic ordering (Borissova and Mustakerov 2017), different linear programming methods (Lewis and Butler 1993; Wu et al. 2007) and Fibonacci search (Wendell 1980; Oliveira and Ferreira 2000). On the contrary, a significant amount of the evolutionary methods is based on NSGA-II and its variants (in 14 papers) (Bechikh et al. 2011, 2013; Pfeiffer et al. 2008; Xin et al. 2018; Nag et al. 2014; Liu et al. 2011; Li and Hu 2023; Cinalli et al. 2015, 2020; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Kadziński and Tomczyk 2017). The NSGA-II variants include preference based R-NSGA-II (Deb and Sundar 2006) and the other variants involve different preference models, such

Fig. 9 MOO methods in the reviewed papers

as I-NOSGA (Balderas et al. 2019) based on interval numbers or NEMO (Branke et al. 2015) based on utility functions. Two papers used SPEA-2 (Hadas and Nahum 2016; Cinalli et al. 2015), two papers SMS-EMOA (Cinalli et al. 2015, 2020) and the other evolutionary methods include IEMO/D_{GD} (Tomczyk and Kadziński 2022) and ASMiGA by Nag et al. (2018).

An overview of the preference aggregation types used in the papers is given in Fig. 10. Type B methods are used in 14 papers (Nag et al. 2014, 2018; Pfeiffer et al. 2008; Xiong et al. 2013; Maharjan and Hanaoka 2018; Subulan et al. 2015; Ahmad

Fig. 10 Preference aggregation types in the reviewed papers

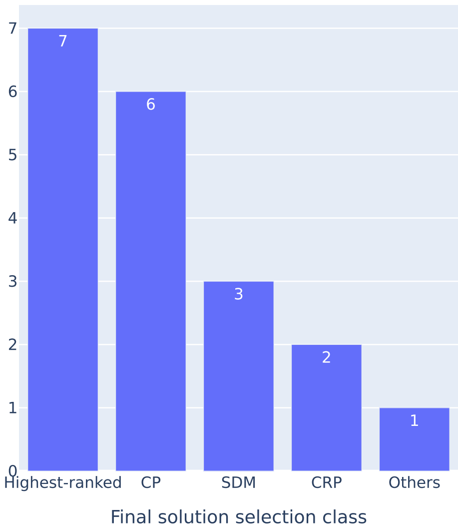


et al. 2022; Chiu et al. 2019; Cinalli et al. 2015, 2020; Tomczyk and Kadziński 2022; Kadziński and Tomczyk 2017; Sakamoto et al. 2021, 2022), type D methods in eight papers (Li and Hu 2023; Bechikh et al. 2011, 2013; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Lewis and Butler 1993) and type C methods in five papers (Borissova and Mustakerov 2017; Hadas and Nahum 2016; Wu et al. 2007; Wendell 1980; Oliveira and Ferreira 2000). Three papers used type A methods (Baharmand et al. 2020; Varas et al. 2020; Baril et al. 2012). In the type D methods, five papers (Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Lewis and Butler 1993) combine types A and C and three papers combine types A and B (Bechikh et al. 2011, 2013; Li and Hu 2023). The most used aggregation operator is the mean, used in 10 papers (Bechikh et al. 2011, 2013; Subulan et al. 2015; Ahmad et al. 2022; Sakamoto et al. 2021, 2022; Xiong et al. 2013; Nag et al. 2014, 2018; Pfeiffer et al. 2008).

Different approaches are used in supporting the group to select the final solution, which we roughly divide to *highest-ranked solution*, according to the collective preference (*CP*), apply a *CRP* and an *SDM selects*. In the highest-ranked solution class, the final solution is selected by conducting a ranking using any type C method. In the CP class, the final solution is selected by exploiting the collective preference, e.g., selecting the solution that is best according to the collective preference. In CRP class, the final solution is selected by conducting a CRP or applying some other type A method. Finally, in the SDM selects, the final solution (or the method to select the solution) is selected by the SDM.

So far, we have mentioned all methods proposed in the literature, but not all of them solved the problem and selected the final solution. Only in 19 out of the 31 papers, the final solution is selected, see Fig. 11. The highest-ranked solution approach is used in seven papers (Balderas et al. 2022; Fernández et al. 2021; Fernández and Olmedo 2013; Baril et al. 2012; Wu et al. 2007; Hadas and Nahum 2016; Lewis and Butler 1993), while in six papers the final solution is selected

Fig. 11 Final solution selection classes in the reviewed papers



according to the collective preference (Maharjan and Hanaoka 2018; Ahmad et al. 2022; Subulan et al. 2015; Chiu et al. 2019; Li and Hu 2023; Fernández and Olmedo 2013). In three papers (Fernández et al. 2022; Borissova and Mustakerov 2017; Fernández and Olmedo 2013), the SDM selects the final solution and in two papers the final solution is selected by applying a CRP (Baharmand et al. 2020; Varas et al. 2020). In others (Cinalli et al. 2015), it was not elaborated by the authors why the final solution was selected.

Some interactive methods (Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Wu et al. 2007) conducted the final solution selection in a post-processing stage after the solution process. While post-processing can be a natural way of applying existing MCGDM methods in solving MOPs, the questions arise on i) when to stop the solution process and ii) how to select the post-processing method? Determining the proper point in time to stop the solution process to be ready to select the final solution via post-processing may prove difficult, as in principle (depending on the problem), in interactive methods, new unseen solutions can be found that may satisfy the DMs better. Moreover, different final solution selection approaches may lead to different solutions for the group, e.g., majority vote or a SDM selection. Depending on the DMs and the group structure, this may be difficult and possibly require group decision support in itself.

In the following sections, we discuss the GDM-MOO methods in some more detail. Since the columns in the upcoming Tables 2, 3 and 4 are similar, we introduce their meanings here. We present the reference and the name of the method (if available) in the first column *'Reference'*. The second column *'Problem type, k'* details the problem type and the number of objective functions k considered. After this, we provide preference information types in the column *'Preference types'*. They have been defined in Sect. 2. We also report whether the MOO method used is either scalarization or evolutionary-based on the column *'Method type'*. Furthermore, we mention the specific methods by name if available.

In the column *'PA'*, we inform what type of preference aggregation is used, as defined in Sect. 2. Additionally, for combined methods, we report the combination. The symbol '*' depicts that the DMs' degrees of importance are considered in some way. We report whether the method or the authors determine the final solution in the column *'FS'* with a ✓and ✗, for selecting the final solution and not selecting the final solution, respectively. Additionally, the final solution selection class is noted. Information on what type of group structure is used is given in the column *'Group structure'*. The different group structures are defined in Sect. 2. In the last column, we indicate the type of DMs and the total number of DMs *'Type of DMs, m'*. Additionally, we report whether the DMs' degree of importance is assumed to be equal or not. Lastly, Sect. 6 considering interactive methods involves Table 4 that has one more column, *'Stopping criterion'*, where we report how the solution process is stopped. The notation '-' means that this information is not mentioned.

The following two sections discuss the GDM-MOO methods and the mentioned information in the tables. They also answer the research questions of what kind of MOPs are solved, how the multiple preferences from several DMs are incorporated and what methods are used to support the group in the final solution selection. The non-interactive, a priori and a posteriori methods are discussed in Sect. 5, and

Table 2 A priori methods: reference and method's name (if available), problem type and number of objective functions k, preference type, MOO method type, preference aggregation type (PA), final solution selection (FS), group structure, type of DMs and number of DMs m

Reference	Problem type, k	Preference types	Method type	PA	FS	Group structure	Type of DMs, m
Maharjan and Hanaoka (2018)	Real-world application, 2	Ranking of objective functions	Scalarization, weighted sum	B	✓, CP	Committee	Equal expert DMs, 4
Subulan et al. (2015)	Real-world inspired problem, 3	Linguistically articulated goals	Goal programming	B*	✓, CP	Team	Non-equal expert DMs, SDM, 4
Ahmad et al. (2022)	Real-world inspired problem, 2	Weights, hesitant values for objective functions	Scalarization, weighted sum	B	✓, CP	Team	Equal unspecified DMs, SDM, 4
W-NSS-GPA Bechikh et al. (2013)	Real-world inspired problem, 2	Reference point, deviation value	Evolutionary, r-NSGA-II	A, B*	✗	Established group	Non-equal unspecified DMs, 10
NSS-GPA Bechikh et al. (2011)	Test problem, 4	Reference point, deviation value	Evolutionary, r-NSGA-II	A, B	✗	Established group	Equal unspecified DMs, 10
Pfeiffer et al. (2008)	Real-world inspired problem, test problems, 2	Reference point, additional parameter	Evolutionary, r-NSGA-II	B	✗	–	Equal unspecified DMs, 2–4
Xiong et al. (2013), Nag et al. (2014)	Test problem, 2	Fuzzy reference point	Evolutionary, NSGA-II	B	✗	–	Non-equal unspecified DMs, 5
Nag et al. (2018)	Test problem, 2	Fuzzy reference point or triangular or Gaussian membership function	Evolutionary, ASMiGA	B	✗	–	Non-equal unspecified DMs, 3
MWS-NSGA Liu et al. (2011)	Test problem, 2	Weights for objective functions	Evolutionary, MWS-NSGA	–	✗	–	Unspecified DMs, 2 or 5

Table 3 A posteriori methods: reference and method's name (if available), problem type and number of objective functions k, preference types, MOO method type, preference aggregation type (PA), final solution selection (FS), group structure, type of DMs and number of DMs m

Reference	Problem type, k	Preference types	Method type	PA	FS	Group structure	Type of DMs, m
Baharmand et al. (2020)	Real-world application, 3	—	Scalarization, augmented ε-constraint	A	✓, CRP	Established group	Equal expert DMs, 4
Borissova and Mustakerov (2017)	Real-world application, 2	Weights or ranking of objective functions	Scalarization, weighted sum or lexicographic ordering	C*	✓, SDM	Team	Non-equal expert DMs, SDM, 4
Hadas and Nahum (2016)	Real-world application, 3	Weights, pairwise comparisons of objective functions, bounds	Evolutionary, SPEA-2	C	✓, Highest-ranked	–	Equal expert DMs, 4
WIN Chiu et al. (2019)	Test problems: 2-3, real-world inspired problem: 6 criteria	Ranking of objective functions	Scalarization, WIN	B*	✓, CP	Team	Non-equal expert DMs, SDM, 5

Table 4 Interactive methods: reference and method's name (if available), problem type and number of objective functions k, preference type, MOO method type, preference aggregation type (PA), final solution selection (FS), stopping criterion, group structure, type of DMs and number of DMs m

Reference	Problem type, k	Preference types	Method type	PA	FS	Stopping criterion	Group structure	Type of DMs, m
Varas et al. (2020)	Real-world application, 2	-	Scalarization, Tchebycheff, ε-constraint	A	✓, CRP	Agreement, no more solutions to compare	-	Equal expert DMs, 2
Li and Hu (2023)	Real-world application, 2	Reference point, additional parameters	Evolutionary, custom NSGA-II	A, B *	✓, CP	Number of iterations	Committee	Non-equal expert DMs, 5
Lewis and Butler (1993)	Real-world inspired problem, 3	Ranking of solutions	Scalarization, SIMOLP, Tchebycheff	A, C	✓, Highest-ranked	Agreement, number of iterations	Established group	Equal student DMs, 4
CI-NSGA-II, CI-SMS-EMOA Cinalli et al. (2015)	Test problems 2-3, real-world inspired problem 2	Pairwise comparison of solutions	Evolutionary, modified NSGA-II	B	✓, Other	Number of iterations	–	Equal student DMs, 30
CI-NSGA-II, CI-SMS-EMOA, CI-SPEA2 Cinalli et al. (2020)	Real-world inspired problem, 2	Pairwise comparison of solutions	Evolutionary, modified NSGA-II	B	✗	Technical stopping criterion	–	Equal student DMs, 30
Wu et al. (2007)	Real-world inspired problem, 3	Most preferred solution	Scalarization, FMOLP, FMOLGP, IFMOLP	C *	✓, Highest-ranked	–	Team	Non-equal unspecified DMs, SDM, 3
CIMO Baril et al. (2012)	Real-world inspired problem, 2	Classification of objective functions	Scalarization, goal programming	A	✓, Highest-ranked	Agreement	Team	Non-equal unspecified DMs, SDM, 3

Table 4 (continued)

Reference	Problem type, k	Preference types	Method type	PA	FS	Stopping criterion	Group structure	Type of DMs, m
Fernández and Olmedo (2013)	Real-world inspired problem, 9	Most preferred solution, weight for objective functions, additional parameters	Evolutionary, NSGA-II	A, C	✓, SDM, highest-ranked, CP	Number of function evaluations	Established group, committee/team	Equal unspecified DMs, SDM, 6
Fernández et al. (2021)	Real-world inspired problem, 9	Most preferred solution, interval weights, additional parameters	Evolutionary, I-NOSGA	A, C	✓, Highest-ranked	Number of function evaluations	Committee	Equal unspecified DMs, 10
Fernández et al. (2022)	Real-world inspired problem, 9	Most preferred solution, interval weights, additional parameters	Evolutionary, I-NOSGA	A, C	✓, SDM selects	Number of function evaluations	Committee, team	Equal unspecified DMs, SDM, 10
Balderas et al. (2022)	Real-world inspired problem, 9	Most preferred solution, interval weights, additional parameters	EMO, I-NOSGA	A, C	✓, Highest-ranked	Number of function evaluations	Established group	Equal unspecified DMs, 10
IEMO/D_{GD} Tomczyk and Kadziński (2022)	Test problems 2-5, real-world inspired problem 3	Pairwise comparison of solutions	Evolutionary, IEMO/D	B	✗	Number of iterations	Committee	Non-equal unspecified DMs, 2–5
NEMO-GROUP Kadziński and Tomczyk (2017)	Test problems 2-7	Pairwise comparison of solutions	Evolutionary, NEMO-GROUP	B	✗	Number of iterations	Committee	Non-equal unspecified DMs, 2–7

Table 4 (continued)

Reference	Problem type, k	Preference types	Method type	PA	FS	Stopping criterion	Group structure	Type of DMs, m
Wendell (1980); Oliveira and Ferreira (2000)	Test problem, 2	-	Fibonacci search	C	✗	Technical stopping criterion	Committee	Equal artificial DMs, 3
Mhab-EC Sakamoto et al. (2021, 2022)	Test problem, 2	Weights for objectives	Evolutionary, -	B	✗	Number of iterations	–	Equal artificial DMs, 300

interactive methods in Sect. 6. The papers are discussed in the following order: first, papers using expert DMs and selecting the final solution, followed by papers with artificial DMs and methods not selecting the final solution. Furthermore, the discussions are grouped by methodological similarities.

5 Non-Interactive Methods

In this section, we discuss non-interactive methods, beginning with a priori methods presented in Table 2. A real-world application in disaster management (Maharjan and Hanaoka 2018) aims to select the locations of temporary logistic hubs. A committee of four expert DMs (from different humanitarian organizations of Nepal) linguistically articulate fuzzy ranking of the objective functions and each ranking is subsequently transformed into a weight vector. These weight vectors from the DMs are aggregated to a collective weight vector using a MCDA method. In this way, using the weighted sum method, the final solution is determined.

A fuzzy linear MOP is formed in Subulan et al. (2015) to solve a real-world inspired problem modeling a supply chain network with three objective functions. A team of three expert DMs (supply chain director, financial advisor and marketing & recycling manager) and an SDM (board chairman) apply goal programming. First, the SDM sets the importance of each DM with weights and then the DMs linguistically articulate their preferences as goals. The preferences are transformed to fuzzy goals and by using a weighted geometric mean operator, a collective preference is formed. Then, the final solution is determined by exploiting the collective preference.

Ahmad et al. (2022) consider a real-world inspired biobjective problem in a product manufacturing system. A team of three DMs and an SDM articulate their preferences as weights for objective functions. Additionally, they provide so-called hesitant values for fuzzy aggregation operators to reflect the imprecision in preference articulation. Different final solutions can be found by utilizing different hesitant fuzzy aggregation operators in a transformed problem formulation solved by a weighted sum based approach.

In the following papers, we focus on describing how the preferences are incorporated and aggregated in the methods as the papers did not solve real-world applications, select the final solution or engage expert DMs. In methods called NSS-GPA and W-NSS-GPA (Bechikh et al. 2011, 2013), each DM is represented by an agent DM and they participate in a predetermined number of rounds of a CRP, facilitated by a moderator agent. The DMs articulate their preferences as reference points and also specify acceptable deviations for each component of the reference point. Additionally, the W-NSS-GPA method utilizes an a priori-defined weight vector depicting the degrees of importance of the DMs. During these rounds, the human DMs communicate through a graphical user interface adjusting their preferences and requesting other DMs to adjust their preferences. The moderator gives feedback to the DMs so they can adjust their preferences to be closer to a collective reference point, which is defined as the mean of the DMs' reference points.[2] After the CRP has finished, the collective reference point is given to R-NSGA-II to generate a set of nondominated solutions reflecting it. The authors do not specify how the final solution is selected from the resulting solutions.

As described in Pfeiffer et al. (2008), the DMs provide their preferences as reference points and a value for an additional parameter that affects the diversity of the generated solutions. The proposed method includes four variations that modify how R-NSGA-II selects surviving population members according to the group's preferences. In this way, the solutions generated by R-NSGA-II are concentrated between the DMs' preferences and spread according to the additional parameter.

Similar methods are presented by Xiong et al. (2013); Nag et al. (2014), where non-equally weighted DMs articulate their preferences using fuzzy reference points in the objective space. In both papers, a modified version of NSGA-II is used to generate a set of collective solutions, from which no final solution is selected. Xiong et al. (2013) formulate the authors formulate a new MOP with two objective functions: a consensus measure and a robustness measure. The consensus measure incorporates both distances to the collective preference and distances among the DMs' preferences, while the robustness measure aims to indicate the solution's ability to handle any changes in the preferences of the DMs. Instead, Nag et al. (2014) include an additional objective function in the original MOP, which combines the mentioned consensus measure with a robustness measure into a single measure defined as a robust consensus measure. The method proposed by Nag et al. (2018) extends the work of Nag et al. (2014) by allowing the DMs to articulate their preferences as Gaussian or triangular membership functions and proposes several versions of the robust consensus measure utilizing various aggregation operators such as weighted conjunction and weighted T-norm operators.

Finally, Liu et al. (2011) proposes a method using a modified version of NSGA-II. The DMs articulate their preferences as weights for objective functions and the method prefers non-dominated solutions closer to the non-dominated solutions corresponding to the weights for objectives. In this way, solutions are found according to multiple preferences. However, there is no preference aggregation and hence the final population members are focused on the regions on the nondominated front that the preferences correspond.

In what follows, we discuss a posteriori methods outlined in Table 3. Baharmand et al. (2020) use data from the 2015 Nepal earthquake to model a problem of determining the placement of temporary relief distribution centers. The three objective functions represent response time, logistical costs and unsatisfied demands. The authors use the augmented ε-constraint method to generate a set of Pareto optimal solutions. Then, a CRP in-person involving a total of eight real humanitarian expert DMs is performed. The DMs are split into two established groups with four DMs each. A moderator (one of the authors) supports the DMs and ensures that everyone clearly understands the decision situation. The DMs are shown Pareto optimal solutions and some visualizations of the solutions and the DMs must converge to a final

[2] A weighted mean for W-NSS-GPA.

solution using group discussions in one hour. The authors provided no further detail on how group discussions and moderator interactions work. In the end, the groups select a different solution as the final solution.

The method presented by Borissova and Mustakerov (2017) aim to determine an optimal wind farm layout with two objective functions of energy production and costs. At the start, the SDM articulates weights for objectives or a ranking of objective functions. The preferences are used to generate different Pareto optimal solutions to be evaluated either by using a weighted sum or lexicographic ordering. Next, the SDM gathers a team of three DMs and defines the degrees of importance of DMs with weighting coefficients. Then, the DMs evaluate the solutions individually without discussing with each other and order the solutions by providing a score value for each of the solutions. Finally, by combining the weights for objectives, the weights for DMs and the score values of solutions to a weighted sum formulation, a final solution is selected.

Hadas and Nahum (2016) re-examine an earlier case study, a real-world problem optimizing public transportation networks. The objective functions are total travel time savings, maintain origin and destination terminals and construction cost. The problem is solved with an evolutionary method (SPEA-2) and a set of non-dominated solutions is generated. The authors distribute an online questionnaire to the four DMs representing authorities and users. In the questionnaire, the DMs are asked to give weights for objective functions, the lower and upper bounds for objective function values and pairwise comparison of objective functions. Three different MCDA ranking methods are utilized to get for each DM three different rankings of the solutions. Then, a voting method, the so-called Borda count method, is used to combine all the individual rankings into a collective ranking. The highest-ranked solution is selected as the final one.

Chiu et al. (2019) solve a real-world inspired MCDA type of problem of selecting the best engine design from four options according to six different criteria and involving five expert DMs in addition to the SDM structured as a team. In addition, several test problems with 2–3 objective functions and two DMs are tested. Note, that the authors assume an approximation of the Pareto front to be found by an evolutionary method or the existence of the Pareto front, before starting the a posteriori GDM process. The method contains three main steps: the DMs rank the objective functions, an SDM determines the degrees of importance of the DMs and forms the collective weights for objectives and finally selects the final solution through distance minimization using the weight induced norm (WIN) method.

6 Interactive Methods

In Table 4, we summarize key aspects of interactive GDM-MOO methods. We start with a real-world application optimizing the wine-harvest schedule with two objective functions by Varas et al. (2020). The objective functions are operational costs and the quality of the harvested grapes. Two expert DMs (a field manager and an oenologist) work daily with wine-harvest operations. The solution process starts by finding a Pareto optimal solution by minimizing the augmented weighted Tchebycheff distance to the ideal vector. This solution is suggested to the DMs and if it satisfies both DMs, the solution process ends. Otherwise, the augmented ϵ-constraint method is used to find Pareto optimal solutions near the solution last shown to the DMs. This approach is iterated until an agreement is reached or no more solutions are found.

Another real-world application, a portfolio optimization problem, is presented by Li and Hu (2023). It has two objective functions, the return and risk of investment. A committee of five expert DMs work in the same company and as they personally know each other, they can evaluate each other's decision making qualities. An evolutionary method is initialized with the DMs articulating their preferences as a reference point and additional parameter values depicting the DMs' evaluation of the other DMs. After this point, the actual involvement of the DMs in the solution process is not explicated in the paper, and it is not clear whether the DMs or the DMs' preference models (value functions) are evaluating the solutions. The interactive solution process iteratively generates a representative set of non-dominated solutions with a modified NSGA-II, from which a subset (selected by clustering) is then pairwise compared. Then, a collective preference is obtained with an automated CRP that utilizes a weighted aggregation of the DMs' preferences. The modified NSGA-II selects solutions according to the collective preference. The solution process ends when a predetermined number of iterations has been conducted. Then, the final solution is selected corresponding to the solution with the highest level of agreement according to a customized function that the authors design.

Lewis and Butler (1993) present an interactive method tested with several four-person established groups of student DMs solving a three objective real-world inspired scheduling problem. The solution process starts with generating a small number of solutions using a linear programming method called SIMOLP or a Tchebycheff function. The DMs conduct group discussions before ranking the solutions. The rankings are aggregated into a collective ranking of the solutions of which the highest-ranked solution is suggested as the final solution. If there is a full agreement among the DMs, this solution is the final solution. Otherwise, the iterative process is repeated until the majority of the DMs wish to end the solution process by selecting the final solution with a majority vote.

Cinalli et al. (2015); Cinalli et al. (2020) incorporate so-called collective intelligence of DMs, in this case, 30 student DMs, when solving a gamified real-world inspired facility location problem with two objective functions. Evolutionary methods called CI-NSGA-II, CI-SMS-EMOA and CI-SPEA2 are proposed, which adjust evolutionary operators to be compatible with multiple DMs. The solution process starts by generating non-dominated solutions and after every predetermined amount of generations, the DMs perform pairwise comparisons of two randomly selected non-dominated solutions. The other non-dominated solutions are evaluated by a Gaussian mixture model.[3] In Cinalli et al. (2015), after seven iterations, the solution process ends. Then, the DMs are shown a solution, which is claimed to be the

[3] It is assumed that all the DMs' pairwise comparisons have approximately a Gaussian distribution.

best solution for the group and it is determined as the final solution. In Cinalli et al. (2020), there are two alternative stopping criteria, convergence and run time of the method. Neither of the stopping criteria consider the goodness of the solutions for the group and the final solution selection is not discussed.

Wu et al. (2007) consider a real-world inspired problem that is modeled as a fuzzy linear MOP solved by a team of two unspecified DMs and an SDM. The method has two stages and in the first stage, each DM finds their most preferred solution by applying a fuzzy linear MOO method of their choosing. Then, the DMs show their most preferred solutions to others via a graphical user interface. The second stage focuses on this set of solutions and uses different MCDA methods to rank them. This involves several phases including the SDM determining the DMs' degrees of importance and the possibility to add new criteria to evaluate the solutions. The final solution is the highest-ranked solution. If the SDM is not satisfied with the final solution, the SDM decides whether to continue the solution process from the first or the second stage.

A real-world inspired problem is solved in a so-called "decentralized environment" setting by Baril et al. (2012). The problem with three objective functions is related to the structure optimization of a two-bar truss. A team of an SDM and two DMs solves different subproblems. The SDM's subproblem has two of the objective functions and one DM considers one objective function while the other DM minimizes constraint violations. The solution process can be described as a system-led CRP, where in a decentralized manner (or in parallel), the SDM and the DMs solve their subproblems. The DMs classify the objective functions in each iteration and the SDM's solution process constrains the DMs' solution processes by shared parameters and auxiliary variables. The solution process is repeated until everyone is satisfied with the final solution or a predetermined number of iterations is met. The authors suggest using AHP to select the highest-ranked solution as the final solution if a full agreement is not otherwise reached.

Fernández and Olmedo (2013), Fernández et al. (2021), Balderas et al. (2022), Fernández et al. (2022) solve a real-world inspired project portfolio optimization problem using interactive evolutionary methods and MCDA based preference modeling. The problem contains 100 different projects with nine objective functions describing the impacts of these projects on different social groups. The papers contain variants of the developed methods for teams or committees in Fernández et al. (2021, 2022) and for established groups in Balderas et al. (2022), or for all of them in Fernández and Olmedo (2013). Before starting the solution process, the DMs first solve the MOP on their own to get their most preferred solution. Then, before the DMs articulate their preferences, a CRP is conducted guided by the SDM (moderator for Balderas et al. (2022)). Additionally, in Fernández et al. (2021, 2022); Balderas et al. (2022), the DMs can choose either multi-criteria ordinal classification or utility function based preference modeling. The preference models classify the found solutions (to satisfy or dissatisfy) the DM. The different preference models (which these methods rely on) are discussed next.

The methods require that the DMs articulate the most preferred solution (found by solving the original MOP), weights for the objective functions and additional parameters for the chosen preference model. As described in Fernández and Olmedo

(2013), each DM sets up an MCDA based preference model and in Fernández et al. (2022), an SDM constructs preference models for each DM. The DMs articulate a so-called limiting boundary, which classifies the solutions to either satisfy or dissatisfy that DM in Fernández et al. (2021). Lastly, in Balderas et al. (2022), each DM classifies six so-called representative solutions, to be either satisfactory or unsatisfactory. It is not elaborated on how these representative solutions are selected or generated. By incorporating the preference models, evolutionary methods such as I-NOSGA generate a set of collective solutions. When to stop the solution process to move to post-processing to select the final solution depends on the satisfaction level of these collective solutions. Otherwise, these methods restart from either conducting an additional CRP or by DMs articulating their preferences. In Fernández and Olmedo (2013), if this predetermined satisfaction level is met, the final solution is selected using an MCDA method or a customized consensus measure function that the authors design. In Fernández et al. (2021, 2022), the SDM specifies the method to select the final solution or restarts the process from the beginning. As reported by Balderas et al. (2022), depending on the number of collective solutions found, either a Borda count or a majority vote selects the final solution.

The next two papers use evolutionary methods assuming an underlying utility function to exist for each DM. Furthermore, the problems solved are mainly test problems with 2–7 unspecified DMs and as the DMs' roles are played by utility functions, we focus describing how the preferences are incorporated and aggregated in the methods.

An evolutionary method called IEMO/D_{GD} is proposed by Tomczyk and Kadziński (2022). It co-evolves two populations, one focusing on approximating the Pareto front, while the other focuses on generating collective solutions. The method creates a collective preference model based on pairwise comparisons of solutions performed by DMs after every predetermined amount of generations. The collective preference model guides the search and is iteratively updated based on the pairwise comparisons. The solution process stops when a predetermined number of iterations have been conducted and the final population is presented as collective solutions. No final solution is selected.

Kadziński and Tomczyk (2017) present an interactive evolutionary method based on NSGA-II adapted for multiple DMs called NEMO-GROUP, containing several variants. After every predetermined amount of generations, the DMs perform pairwise comparisons of randomly selected non-dominated solutions. Based on the pairwise comparisons, different NEMO-GROUP variants construct different collective preference models. The collective preference model used evaluates the individuals for inclusion in the next population. The solution process stops when a predetermined number of iterations has been conducted and no final solution is selected.

In addition to the discussed methods, there are other methods developed for some specific purpose, tested with biobjective test problems and artificial DMs. Methods of Wendell (1980); Oliveira and Ferreira (2000) aim to find a set of collective solutions for a committee of three DMs in solving biobjective problems with some additional mathematical assumptions. The DMs are assumed to have a utility function to model their preferences and in each iteration, the insight provided by each utility function constraints the decision space until the set of collective solutions is found. However, the authors do not discuss how to select the final solution from the collective solutions if it contains multiple solutions.

An agent-based evolutionary[4] MOO method is proposed by Sakamoto et al. (2021, 2022). The main drive of using agents is to avoid direct communication of the DMs. The method is tested with a biobjective test problem, where the authors randomly generate weights for objective functions as the DMs' preferences (represented by an agent). However, it is not elaborated on how many agents are representing a DM's preferences, but each agent acts as a member of the evolutionary population. Therefore, we report the population size as the number of artificial DMs. The evolutionary method generates so-called consensus solutions and the final solution is not selected.

7 Discussion

In this section, we discuss the results of the systematic literature review on GDM-MOO methods for solving GDM-MOPs. The review resulted in the main classes of non-interactive and interactive methods and explicated with structuring the field according to types of DMs, types of MOPs, types of group structures, GDM-MOO (and MOO) method types, types of preference information, ways to aggregate preferences and detected approaches to select the final solution to synthesize the current state-of-the-art.

The discussion is structured as follows. In Sect. 7.1, we identify key issues in the current literature and summarize how they are considered in the current literature in Sect. 7.2. In Sect. 7.3, we give guidance on how to select a suitable GDM-MOO method for solving real-world GDM-MOPs.

7.1 Issues in the GDM-MOO Literature

As reported, a plethora of GDM-MOO methods exist with diverse theoretical backgrounds and approaches. This variety highlights the importance of a systematic literature review, synthesizing the current state-of-the-art. It is useful that there are a wide range of different methods as from the variety emerges the chance to apply them in solving various kinds of real-world applications. However, the variety has resulted in a fluctuation in the fundamental concepts such as (i) the interpretation of what solving a GDM-MOP means and (ii) what is a realistic decision situation to test and validate the methods in.

To combat these two main themes, we identify the following key issues (KI) that need addressing when proposing GDM-MOO methods:

[4] The evolutionary method used is not specified.

- **Validating the practical applicability of the proposed methods.**
  - KI1: Validating the methods by solving real-world applications.
  - KI2: Selecting the final solution.
  - KI3: Validating the acceptability of the final solution.
  - KI4: Validating the acceptability of the solution process.
- **Specifying the aspect of the group.**
  - KI5: Specifying who the DMs are.
  - KI6: Specifying how the group is structured.
  - KI7: Specifying whether the DMs communicate in-person, online or not at all.
  - KI8: Specifying if and when the DMs know about the preferences of the other DMs or about the collective preference.

When these specifications have been made, the reader can conclude whether the proposed method is relevant for their needs. As mentioned, as a large portion of the proposed methods ignore or do not specify these key issues, some actions have to be taken. However, it is not obvious, how to build a proper test setting to validate the GDM-MOO method's real-world applicability. Next, we discuss a few examples on what kind of test settings could be used in the validation of the proposed methods.

What is the test setting utilized to validate the applicability of the proposed method?

To be able to assess real-world applicability, a meaningful test setting should involve validating the proposed methods by solving real-world applications (KI1), where solving the problem means selecting the final solution for the group (KI2). Otherwise, as mentioned, it is not clear what decision is to be put into practice. It would be desirable to validate the acceptance of the final solution (KI3) and similarly, validate the acceptance of the solution process (KI4). As the final solution has to be acceptable to the group (or at least to the group members responsible for implementing the solution), otherwise, the solution may not be implemented at all. One way to achieve this could be by simply asking the DMs a few questions after the solution process. Such questions could for example be *I think that the final solution is the best one for the group*, *I accept the final solution* or *I was able to articulate my preferences as I wanted* (to be answered on a Likert scale) (modified from Afsar et al. (2023) for multiple DMs).

Furthermore, it is still relevant to indicate who the DMs are (KI5) and how the group is structured (KI6). This should not be an issue in real-world applications that involve a group of expert DMs as they have their roles and different degrees of importance in reality. However, the solution process is affected by how the DMs in the group operate. This involves how the communication between the DMs function (KI7) and whether the DMs know about the preferences of others or about the collective preference (KI8). By following this ideal test setting, the proposed method would be validated regarding solving the specific real-world application among with the specifications of KI2–KI8. How to define the ideal test setting for a more general purpose such as to be able to compare different GDM-MOO methods in solving various GDM-MOPs is a clear future research direction.

It is not always possible to have a real-world application or expert DMs. Therefore, a realistic test setting involves solving a real-world inspired problem (KI1) with selecting the final solution (KI2). While the acceptability of the final solution and the solution process may be less meaningful (as it is not a real problem), KI3–K4 should be still questioned from the DMs and reported in the papers. Especially, the understandability aspect of the solution process is interesting when solving real-world inspired problems. Moreover, if it is not possible to involve expert DMs, testing the methods with well-informed (of the problem) students acting as DMs and reporting the specifications of KI5–KI8, is preferred to not having any validation with humans. Whether the proposed method is a proof-of-concept or with a more general purpose, testing with this kind of realistic test setting would be useful.

Finally, the practical applicability of the methods cannot be justified when tested with bench-marking test problems that involve no human elements, where KI2–KI8 are unspecified. The reader can not imagine how would the proposed method work on their specific problem setting. Therefore, the realism of the whole solution process can be questioned and with that, the proposed method's validity for solving real-world applications. However, solving test problems may be useful for other purposes such as comparing different methods.

7.2 Summary of How the Key Issues are Considered in the Current Literature

We have seen the need to pay attention to the key issues listed in Sect. 7.1 because of the shortcomings observed in the literature. Note that we do not classify these papers to the described test settings as i) these test settings are for descriptive purposes only and future research has to be conducted to define them more appropriately and ii) most of the literature does not fulfill the key issues to be classified in any of them. In what follows, we summarize the number of papers where each of the key issues has been considered, listed in Table 5. Moreover, we suggest some approaches to better fulfill the key issues when proposing new methods.

As mentioned, in six papers (Baharmand et al. 2020; Li and Hu 2023; Varas et al. 2020; Borissova and Mustakerov 2017; Maharjan and Hanaoka 2018) real-world applications (KI1) are solved and most of the key issues are considered (expect KI4). The only paper where all key issues are specified was by Lewis and Butler (1993), with the caveat of solving a real-world inspired problem with students playing the role of the DMs. As the current literature has focused on solving test problems or

Table 5 Key issues identified and number of papers considering them

Key issues	# of papers
KI1	6
KI2	19
KI3	8
KI4	1
KI5	15
KI6	23
KI7	11
KI8	11

real-world inspired problems without specifying KI2–KI7, future research needs to address this.

In 19 papers out of 31 reviewed ones, a final solution is selected (KI2). Most commonly the final solution is selected as the highest-ranked solution or by exploiting the collective preference. While following the majority's opinion is a well-known approach in GDM, the literature raises some concern in following the majority's opinion as it overrides the minority's opinions, see e.g., (Fernández and Olmedo 2013; Lootsma et al. 1998; Fernandez and Olmedo 2005).

The acceptance of the final solution is validated (KI3) in eight papers (Ahmad et al. 2022; Lewis and Butler 1993; Hadas and Nahum 2016; Li and Hu 2023; Fernández et al. 2021, 2022; Borissova and Mustakerov 2017; Baharmand et al. 2020), out of 31. Most commonly the validation happened based on a consensus measurement (Li and Hu 2023; Fernández et al. 2021, 2022) or comparing the final solution with the preferences of the DMs (Borissova and Mustakerov 2017; Baharmand et al. 2020; Ahmad et al. 2022). Only in two papers (Lewis and Butler 1993; Hadas and Nahum 2016) the DMs were asked (in a questionnaire after the solution process) about their acceptance of the final solution. In interactive MOO methods with a single DM, it can be natural to assume that if a DM stops iterating when finding a satisfying solution, they would also be satisfied with the solution process. Regarding GDM-MOO, as there are multiple DMs, the same assumption is harder to make. However, in any case, only in three papers (Varas et al. 2020; Lewis and Butler 1993; Baril et al. 2012), a stopping criterion based on the DMs' satisfaction is considered. The acceptance of the solution process (KI4) was considered only by Lewis and Butler (1993): the authors asked the opinions of the DMs about the solution process with a question *"The decision process and potential solution sets helped my group identify a better compromise solution than we could have found without using it."*.

Regarding the understandability of the solution process, many methods require the DMs to either articulate complex preference information or be able to specify the method's inner parameters. This is questionable since DMs should be domain experts, not experts of the methods. Therefore, methods should be developed so that DMs can understand how their preferences affect the solution process to learn about the problem and their preferences.

When thinking of validating the proposed methods, in the current literature, the type of DMs is specified (KI5) in 15 papers (Li and Hu 2023; Borissova and Mustakerov 2017; Maharjan and Hanaoka 2018; Baharmand et al. 2020; Hadas and Nahum 2016; Varas et al. 2020; Subulan et al. 2015; Sakamoto et al. 2021, 2022; Wendell 1980; Oliveira and Ferreira 2000; Cinalli et al. 2015, 2020; Lewis and Butler 1993; Chiu et al. 2019) and the group structure (KI6) in 23 papers (Li and Hu 2023; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Kadziński and Tomczyk 2017; Tomczyk and Kadziński 2022; Wendell 1980; Oliveira and Ferreira 2000; Maharjan and Hanaoka 2018; Wu et al. 2007; Baril et al. 2012; Borissova and Mustakerov 2017; Chiu et al. 2019; Ahmad et al. 2022; Subulan et al. 2015; Lewis and Butler 1993; Balderas et al. 2022; Baharmand et al. 2020; Bechikh et al. 2011, 2013). Most of the literature did not explicitly refer to the group structures such as suggested by Marakas (2003) or specify all the related aspects regarding how the group is structured and how it operates during the solution process. The details on how the group operates during the solution process and whether they know about other DMs' preferences (KI7 and KI8) are reported only in 11 papers (Varas et al. 2020; Bechikh et al. 2011, 2013; Lewis and Butler 1993; Fernández and Olmedo 2013; Fernández et al. 2021, 2022; Balderas et al. 2022; Wu et al. 2007; Maharjan and Hanaoka 2018; Baril et al. 2012). Because of the large amount of papers not specifying the KI5–KI8, we can conclude that much of the behavior of the DMs or how the group functions in GDM may have been ignored in the current literature.

We have discussed how the identified key issues have been considered in the literature. When the mentioned specifications are made, it is possible to provide some guidance on how to select a suitable method for a specific real-world GDM-MOP.

7.3 How to Select a GDM-MOO Method for a Real-World Problem?

Next, we suggest some leading questions to guide the reader on how to select a GDM-MOO method for real-world GDM-MOPs:

1. Who are the DMs, and what is their expertise?
2. How is the group structured?
3. Does the group have any preferences on how they wish to operate during the solution process?
4. What are the specifics of the MOP?

The first aspect in selecting a suitable GDM-MOO method is to consider the DMs in question and their expertise in the problem domain. What kind of preference information they can provide, and how much time and effort are they willing to devote to the solution process? Specifically, do they have time to apply an interactive method or is it mandatory to use a non-interactive method.

The second aspect is to consider the structure of the group, including the KI5–KI8. This includes the roles of the DMs in the solution process, whether the group communicates in-person, online or not at all. As DMs' behavior affects the solution process it should be decided whether the DMs should know about each others' preferences or about the collective preference. If so, at which times during the solution process should this information be shared? In addition, how should the moderator act during the solution process? Moreover, when the group has an SDM, how the SDM is supposed to operate e.g., in Fernández et al. (2022) the SDM is supposed to behave "democratically" during the solution process.

The third aspect is to consider how the group would like to solve the problem. The group's preferences on how they would like to operate during the solution process have not received much attention in the reviewed literature. However, for example in Sakamoto et al. (2021, 2022) it was assumed that the group prefers to work anonymously. Furthermore, the future method development should consider how the group is supported in the GDM process. The various aspects related to human behavior (e.g., increase in cognitive load and interpersonal conflicts among DMs) when solving real-world GDM-MOPs cannot be ignored.

Finally, the specifics of the MOP to be solved do matter: how many objective functions and decision variables it contains, what is the computational complexity, etc. The selected GDM-MOO method must be appropriate for these requirements. Moreover, the current literature begins with the assumption that the problem has been already formulated, and it is implicitly assumed that the DMs agree upon the problem formulation involving decision variables, objective functions and constraints to consider. However, there may be problems where all the DMs do not share the same set of objective functions or decision variables, also mentioned by Fernández et al. (2021).

Because of the challenges with the key issues discussed, the practical applicability of the methods proposed in the literature is not always clear. Additionally, because of the lack of existing GDM-MOO methods overall, there may not exist a suitable GDM-MOO method for many real-world GDM-MOPs. This is a call to action to develop new GDM-MOO methods considering the key issues described earlier.

8 Conclusions

In the conducted systematic literature review we examined the state-of-the-art of GDM-MOO methods published in 31 papers. We propose novel classifications and key issues to consider when developing GDM-MOO methods suitable for real-world applications with expert DMs. The main contribution is in the synthesis of the state-of-the-art GDM-MOO methods via explicating the key concepts classified according to the role of the DMs in the solution process. Furthermore, we described the main approaches in GDM-MOO methods used to solve GDM-MOPs and explained the solution processes of the individual methods. Additionally, we described how and when the DMs articulate their preferences, how different ways to aggregate preferences are utilized in the methods and how the group is supported in selecting the final solution.

In conclusion, without testing the proposed methods with realistic test settings (which consider the key issues noted), many important aspects can remain unnoticed or ignored. Thus, it is recommendable to validate new methods with real-world applications or with realistic test settings reporting the specifications to key issues.

We highlight four future research directions. First, developing GDM-MOO methods that i) consider and ii) mitigate the negative effects of aspects related to human behavior in GDM. Second, exploring and testing different ways to aggregate preferences in the context of MOO, where the DMs' preferences may change e.g., due to learning about the problem. Third, paying attention to important viewpoints relevant for practical applicability in the proposed interactive methods. Examples of them include when to stop the solution process and how to select the final solution? Fourth, developing means to effectively test and validate GDM-MOO methods.

In the systematic literature review over 200 papers were identified via the searches. However, a limited number of 31 papers met the inclusion criteria and were included in the analysis. This was due to the focus of this paper on GDM-MOO methods and due to the lack of research on these methods. GDM-MOO is still a new research area involving a wide range of approaches developed with different viewpoints and problems in mind. Future research and method development are needed to address the needs of real-world applications in the method development of GDM-MOO methods.

Acknowledgements This research was supported by the Academy of Finland (grant number 322221 and 352784). The research is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, jyu.fi/demo) of the University of Jyväskylä. This research has received funding partly from the European Union – NextGenerationEU instrument and is funded by the Research Council of Finland under grant number 352784.

Funding Open Access funding provided by University of Jyväskylä (JYU).

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Afsar B, Miettinen K, Ruiz F (2021) Assessing the performance of interactive multiobjective optimization methods: a survey. ACM Comput Surv 54(4):1–27

Afsar B, Silvennoinen J, Misitano G, Ruiz F, Ruiz AB, Miettinen K (2023) Designing empirical experiments to compare interactive multiobjective optimization methods. J Oper Res Soc 74(11):2327–2338

Ahmad F, Adhami AY, John B, Reza A (2022) A novel approach for the solution of multiobjective optimization problem using hesitant fuzzy aggregation operator. RAIRO-Oper Res 56(1):275–292

Bäck T (1996) Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming. Genetic Algorithms. Oxford University Press, New York

Baharmand H, Comes T, Lauras M (2020) Supporting group decision makers to locate temporary relief distribution centres after sudden-onset disasters: A case study of the 2015 Nepal earthquake. Int J Disaster Risk Reduc 45:101455

Balderas F, Fernández E, Gómez-Santillán C, Rangel-Valdez N, Cruz L (2019) An interval-based approach for evolutionary multi-objective optimization of project portfolios. Int J Inf Technol Decision Making 18(04):1317–1358

Balderas F, Fernández E, Cruz-Reyes L, Gómez-Santillán C, Rangel-Valdez N (2022) Solving group multi-objective optimization problems by optimizing consensus through multi-criteria ordinal classification. Eur J Oper Res 297(3):1014–1029

Baril C, Yacout S, Clément B (2012) An interactive multi-objective algorithm for decentralized decision making in product design. Optim Eng 13(1):121–150

Bechikh S, Said LB, Ghédira K (2013) Group preference-based evolutionary multi-objective optimization with non-equally important decision makers: application to the portfolio selection problem. Int J Comput Inf Syst Ind Manag Appl 5(1):278–288

Bechikh S, Said LB, Ghédira K (2011) Negotiating decision makers' reference points for group preference-based evolutionary multi-objective optimization. In: 2011 11th International Conference on Hybrid Intelligent Systems (HIS), pp. 377–382. IEEE
Belton V, Stewart T (2003) Multiple criteria decision analysis: an integrated approach. Kluwer Academic Publishers, Boston
Belton V, Branke J, Eskelinen P, Greco S, Molina J, Ruiz F, Słowiński R (2008) Interactive multiobjective optimization from a learning perspective. In: Branke J, Deb K, Miettinen K, Słowiński R (eds) Multiobjective optimization: interactive and evolutionary approaches. Springer, Berlin, pp 405–433
Boix-Cots D, Pardo-Bosch F, Alvarez PP (2023) A systematic review on multi-criteria group decision-making methods based on weights: analysis and classification scheme. Inf Fusion 96:16–36
Borissova D, Mustakerov I (2017) A two-stage placement algorithm with multi-objective optimization and group decision making. Cybern Inform Technol 17(1):87–103
Branke J, Greco S, Słowiński R, Zielniewicz P (2015) Learning value functions in interactive evolutionary multiobjective optimization. IEEE Trans Evol Comput 19(1):88–102
Cabrerizo FJ, Moreno JM, Pérez IJ, Herrera-Viedma E (2010) Analyzing consensus approaches in fuzzy group decision making: advantages and drawbacks. Soft Comput 14(5):451–463
Chao X, Dong Y, Kou G, Peng Y (2022) How to determine the consensus threshold in group decision making: a method based on efficiency benchmark using benefit and cost insight. Ann Oper Res 316(1):143–177
Charnes A, Cooper WW (1961) Management models and industrial applications of linear programming. John Wiley & Sons, New York
Charnes A, Cooper WW (1977) Goal programming and multiple objective optimizations: Part 1. Eur J Oper Res 1(1):39–54
Chen S-J, Hwang C-L (1992) Fuzzy multiple attribute decision making methods. Springer, Berlin
Chiu W-Y, Manoharan SH, Huang T-Y (2019) Weight induced norm approach to group decision making for multiobjective optimization problems in systems engineering. IEEE Syst J 14(2):1580–1591
Cinalli D, Martí L, Sanchez-Pi N, Garcia ACB (2015) Integrating collective intelligence into evolutionary multi-objective algorithms: interactive preferences. In: 2015 Latin America Congress on Computational Intelligence (LA-CCI), pp. 1–6. IEEE
Cinalli D, Martí L, Sanchez-Pi N, Garcia ACB (2020) Extending collective intelligence evolutionary algorithms: a facility location problem application. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE
Coello CAC (2000) Handling preferences in evolutionary multiobjective optimization: a survey. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00, vol. 1, pp. 30–37. IEEE
Coello CAC, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems. Springer, Boston
Deb K (2001) Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, Chichester
Deb K (2008) Introduction to evolutionary multiobjective optimization. In: Branke J, Deb K, Miettinen K, Słowiński R (eds) Multiobjective optimization: interactive and evolutionary approaches. Springer, Berlin, pp 59–96
Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
Deb K, Sundar J (2006) Reference point based multi-objective optimization using evolutionary algorithms. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 635–642
Fernández E, Olmedo R (2005) An agent model based on ideas of concordance and discordance for group ranking problems. Decis Support Syst 39(3):429–443
Fernández E, Olmedo R (2013) An outranking-based general approach to solving group multi-objective optimization problems. Eur J Oper Res 225(3):497–506
Fernández E, Rangel-Valdez N, Cruz-Reyes L, Gomez-Santillan C (2021) A new approach to group multi-objective optimization under imperfect information and its application to project portfolio optimization. Appl Sci 11(10):4575
Fernández E, Gómez-Santillán C, Rangel-Valdez N, Cruz-Reyes L (2022) Group multi-objective optimization under imprecision and uncertainty using a novel interval outranking approach. Group Decis Negot 31(5):945–994
Figueira J, Mousseau V, Roy B (2005) ELECTRE methods. In: Figueira J, Greco S, Ehrogott M (eds) Multiple criteria decision analysis: state of the art surveys. Springer, New York, pp 133–153

Hadas Y, Nahum OE (2016) Urban bus network of priority lanes: a combined multi-objective, multi-criteria and group decision-making approach. Transp Policy 52:186–196
Haimes YY, Lasdon LS, Wismer DA (1971) On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans Syst Man Cybern 3:296–297
Harvey JB (1974) The Abilene paradox: the management of agreement. Organ Dyn 3:63–80
Herrera F, Herrera-Viedma E, Verdegay JL (1995) A sequential selection process in group decision making with a linguistic assessment approach. Inf Sci 85(4):223–239
Herrera F, Herrera-Viedma E, Verdegay JL (1996) A model of consensus in group decision making under linguistic assessments. Fuzzy Sets Syst 78(1):73–87
Herrera-Viedma E, Cabrerizo FJ, Kacprzyk J, Pedrycz W (2014) A review of soft consensus models in a fuzzy environment. Inform Fusion 17:4–13
Hwang C-L, Yoon K (1992) Multiple attribute decision making: methods and applications: a state-of-the-art survey, Berlin
Hwang C-L, Lin M (1987) Group decision making under multiple criteria: methods and applications. Springer, Berlin
Hwang C-L, Masud ASM (1979) Multiple objective decision making – methods and applications: a state-of-the-art survey. Springer, Berlin
Jacquet-Lagreze E, Siskos Y (2001) Preference disaggregation: 20 years of MCDA experience. Eur J Oper Res 130(2):233–245
Jelassi T, Kersten G, Zionts S (1990) An introduction to group decision and negotiation support. In: Bana e Costa, CA (ed) Readings in multiple criteria decision aid, pp. 537–568. Springer, Berlin
Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput 9(1):3–12
Kadziński M, Tomczyk MK (2017) Interactive evolutionary multiple objective optimization for group decision incorporating value-based preference disaggregation methods. Group Decis Negot 26(6):1215–1240
Keeney RL, Raiffa H, Meyer RF (1993) Decisions with multiple objectives: preferences and value trade-offs. Wiley, New York
Kilgour DM, Eden C (2010) Handbook of group decision and negotiation. Springer, Dordrecht
Labella Á, Liu Y, Rodríguez RM, Martínez L (2018) Analyzing the performance of classical consensus models in large scale group decision making: a comparative study. Applied Soft Computing 67, 677–690
Laengle S, Modak NM, Merigo JM, Zurita G (2018) Twenty-five years of group decision and negotiation: a bibliometric overview. Group Decis Negot 27:505–542
Lewis HS, Butler TW (1993) An interactive framework for multi-person, multiobjective decisions. Decis Sci 24(1):1–22
Li D, Hu S (2023) Adaptive consensus reaching process with dynamic weights and minimum adjustments for group interactive portfolio optimization. Comput Indus Eng 183:109491
Liu G, Wu G, Zheng T, Ling Q (2011) Integrating preference based weighted sum into evolutionary multi-objective optimization. In: 2011 Seventh International Conference on Natural Computation, pp. 1251–1255. IEEE
Lootsma FA, Ramanathan R, Schuijt H (1998) Fairness and equity via concepts of multi-criteria decision analysis. In: Trends in multicriteria decision making: Proceedings of the 13th International Conference on Multiple Criteria Decision Making, pp. 215–226. Springer, Cape Town
Lu J, Ruan D (2007) Multi-objective group decision making: methods, software and applications with fuzzy set techniques. Imperial College Press, London
Maharjan R, Hanaoka S (2018) A multi-actor multi-objective optimization approach for locating temporary logistics hubs during disaster response. J Human Log Supply Chain Manag 8(1):2–21
Marakas GM (2003) Decision support systems in the 21st century. Prentice Hall, New Jersey
Miettinen K (1999) Nonlinear multiobjective optimization. Kluwer Academic Publishers, Boston
Miettinen K (2008) Introduction to multiobjective optimization: noninteractive approaches. In: Branke J, Deb K, Miettinen K, Słowiński R (eds) Multiobjective optimization: interactive and evolutionary approaches. Springer, Berlin, pp 1–26
Miettinen K, Mäkelä MM (2002) On scalarizing functions in multiobjective optimization. OR Spectrum 24:193–213
Miettinen K, Ruiz F, Wierzbicki AP (2008) Introduction to multiobjective optimization: interactive approaches. In: Branke J, Deb K, Miettinen K, Słowiński R (eds) Multiobjective optimization: interactive and evolutionary approaches. Springer, Berlin, pp 27–57

Miettinen K, Hakanen J, Podkopaev D (2016) Interactive nonlinear multiobjective optimization methods. In: Greco S, Ehrgott M, Figueira JR (eds) Multiple criteria decision analysis: state of the art surveys. Springer, New York, pp 927–976
Nag K, Pal T, Mudi RK, Pal NR (2018) Robust multiobjective optimization with robust consensus. IEEE Trans Fuzzy Syst 26(6):3743–3754
Nag K, Pal T, Pal NR (2014) Robust consensus: a new measure for multicriteria robust group decision making problems using evolutionary approach. In: International Conference on Artificial Intelligence and Soft Computing, pp. 384–394. Springer
Nakayama H (1995) Aspiration level approach to interactive multi-objective programming and its applications. In: Pardalos PM, Siskos YCZ (eds) Advances inmulticriteria analysis, pp. 147–174. Kluwer Academic Publishers, Dordrecht
Oliveira SLC, Ferreira PAV (2000) Bi-objective optimisation with multiple decision-makers: a convex approach to attain majority solutions. J Oper Res Soc 51(3):333–340
Öztürk M, Tsoukiàs A, Vincke P (2005) Preference modelling. In: Figueira J, Greco S, Ehrogott M (eds) Multiple criteria decision analysis: state of the art surveys. Springer, New York, pp 27–59
Page MJ, McKenzie JE, Bossuyt PM, Boutron I, Hoffmann TC, Mulrow CD, Shamseer L, Tetzlaff JM, Akl EA, Brennan SE, Chou R, Glanville J, Grimshaw JM, Hróbjartsson A, Lalu MM, Li T, Loder EW, Mayo-Wilson E, McDonald S, McGuinness LA, A, SL, Thomas J, Tricco AC, Welch VA, Whiting P, Moher D (2021) The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. International Journal of Surgery 88, 105906
Palomares I, Estrella FJ, Martínez L, Herrera F (2014) Consensus under a fuzzy context: taxonomy, analysis framework AFRYCA and experimental case of study. Inform Fusion 20:252–271
Pfeiffer J, Golle U, Rothlauf F (2008) Reference point based multi-objective evolutionary algorithms for group decisions. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 697–704
Raiffa H, Richardson J, Metcalfe D (2002) Negotiation analysis: the science and art of collaborative decision making. Harvard University Press, London
Rezaei J (2015) Best-worst multi-criteria decision-making method. Omega 53:49–57
Roy R (1971) Problems and methods with multiple objective functions. Math Program 1:239–266
Roy B (1996) Multicriteria methodology for decision aiding. Springer, Boston
Saaty TL (1980) The analytic hierarchy process. McGraw-Hill, New York
Saint S, Lawson JR (1994) Rules for reaching consensus: a modern approach to decision making. Pfeiffer, Amsterdam
Sakamoto H, Nakamoto K, Ohnishi K (2022) Evolutionary computation system solving group decision making multiobjective problems for human groups. J Adv Comput Intell Intell Inform 26(2):196–205
Sakamoto H, Nakamoto K, Ohnishi K (2021) Acquiring consensus solutions by multi-human-agent-based evolutionary computation. In: 2021 5th IEEE International Conference on Cybernetics (CYBCONF), pp. 12–18. IEEE
Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization. Academic Press, Orlando
Steiner ID (1972) Group process and productivity. Academic Press, New York
Steuer RE (1986) Multiple criteria optimization: theory, computation, and application. Wiley, New York
Subulan K, Taşan AS, Baykasoğlu A (2015) A fuzzy goal programming model to strategic planning problem of a lead/acid battery closed-loop supply chain. J Manuf Syst 37:243–264
Tomczyk MK, Kadziński M (2022) Interactive co-evolutionary multiple objective optimization algorithms for finding consensus solutions for a group of decision makers. Inf Sci 616:157–181
Varas M, Basso F, Maturana S, Osorio D, Pezoa R (2020) A multi-objective approach for supporting wine grape harvest operations. Comput Ind Eng 145:106497
Wang H, Olhofer M, Jin Y (2017) A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. Complex Intell Syst 3:233–245
Wang X, Xu Z, Su S-F, Zhou W (2021) A comprehensive bibliometric analysis of uncertain group decision making from 1980 to 2019. Inf Sci 547:328–353
Wendell RE (1980) Multiple objective mathematical programming with respect to multiple decision-makers. Oper Res 28(5):1100–1111
Wierzbicki AP (1982) A mathematical basis for satisficing decision making. Math Model 3(5):391–405
Wu F, Lu J, Zhang G, Ruan D (2007) The development of a fuzzy multi-objective group decision support system. In: 2007 IEEE International Fuzzy Systems Conference, pp. 1–6. IEEE

Xin B, Chen L, Chen J, Ishibuchi H, Hirota K, Liu B (2018) Interactive multiobjective optimization: a review of the state-of-the-art. IEEE Access 6:41256–41279

Xiong J, Tan X, Yang K-W, Chen Y-W (2013) Fuzzy group decision making for multiobjective problems: tradeoff between consensus and robustness. J Appl Math 2013:657978