Juuso Pajasmaa

Group decision making in multiobjective optimization: A systematic literature review

Master's Thesis in Information Technology

June 7, 2023

University of Jyväskylä

Faculty of Information Technology

Author: Juuso Pajasmaa

Contact information: juuso.p.pajasmaa@student.jyu.fi

Supervisors: Kaisa Miettinen, and Johanna Silvennoinen

Title: Group decision making in multiobjective optimization: A systematic literature review

Työn nimi: Ryhmäpäätöksenteko monitavoiteoptimoinnissa: Systemaattinen kirjallisuuskatsaus

Project: Master's Thesis

Study line: Tietotekniikka

Page count: 85+0

Abstract: This thesis deals with a systematic literature review considering the amalgamation of group decision making and multiobjective optimization. The literature review contains an introduction of the relevant aspects of multiobjective optimization and group decision making, combining these two (mostly separately considered) research topics into another research topic called, group decision making in multiobjective optimization. The thesis answers the question what the state of the art of group decision making in multiobjective optimization is. The thesis classifies the methods found in the literature according to the role the decision makers play in the multiobjective optimization process and presents a classification of the ways to handle multiple preferences (from several decision makers) in multiobjective optimization methods. The thesis discusses the results that are found, and proposes seven desirable properties that should be considered when developing multiobjective optimization methods for group decision making.

Keywords: multiobjective optimization, group decision making, consensus, voting, negotiation

Suomenkielinen tiivistelmä: Tässä tutkielmassa suoritetaan systemaattinen kirjallisuuskatsaus ryhmäpäätöksenteon ja monitavoiteoptimoinnin yhdistelmälle. Kirjallisuuskatsaus sisältää perehdytyksen sekä monitavoiteoptimointiin, että ryhmäpäätöksentekoon yhdistäen nämä kaksi aiemmin suurimmaksi osaksi erillään kehitettyä tutkimusaihetta yhdeksi tutkimusaiheeksi, jota kutsutaan termillä ryhmäpäätöksenteko monitavoiteoptimoinnissa. Tutkielma vastaa siihen, mikä on kyseisen tutkimusalan nykytila kirjallisuudessa. Tutkielma jaottelee löydetyt metodit päätöksentekijöiden roolin mukaan monitavoiteoptimointiprosessissa ja esittää luokittelun tavoista ottaa huomioon päätöksentekijöiden preferenssi-informaatio monitavoiteoptimointimenetelmissä. Tutkielma käsittelee löydettyjä tuloksia ja esittää seitsemän tavoiteltavaa ominaisuutta, jotka pitäisi ottaa huomioon, kun kehitetään monitavoiteoptimointimenetelmiä ryhmäpäätöksentekoon.

Avainsanat: monitavoiteoptimointi, ryhmäpäätöksenteko, konsensus, äänestys, neuvottelu

List of Figures

List of Tables

Contents

1 Introduction

Decision making is familiar to every one of us from everyday decision making problems and it involves selecting between alternative decisions. Often, people aim to make an optimal decision. The optimal decision is often related to the objective of the person making the decision. Optimization refers to identifying the most efficient way of achieving an objective and the optimal decision. Optimization problems are often considered to have a single objective e.g. maximizing profit in a business, minimizing energy consumption for a building or minimizing the travel time on a transportation truck. In such problems, there exists a single optimal solution.

However, this is rarely the case in real-world problems where there are multiple conflicting objectives, such as maximizing profit, but minimizing risks in portfolio optimization or maximizing the speed of the car, while minimizing fuel usage and minimizing the price of the car. The objectives are often conflicting, e.g., a faster car consumes more fuel. Hence, the conflicting objectives bring up the concept of trade-offs, to gain on one objective, one must allow some other objective get worse [52]. For example, to gain a smaller risk in the investment, the profit objective must be allowed to get worse. These kind of problems with multiple conflicting objectives are known as multiple criteria decision making (MCDM) problems [25, 84]. Depending on the properties of the feasible solutions, these types of problems can be segregated into multiattribute decision analysis (MCDA) and multiobjective optimization (MOO) [52]. In MCDA, the set of feasible solutions is explicitly predetermined, discrete and finite [38, 52]. Examples of this type of problems are selecting which car to purchase or determining the locations of power plants.

In MOO, the set of possible solutions is not explicitly known in advance. Instead, the set of solutions is only implicitly known via functions of decision variables, and the set of solutions can contain an infinite number of solutions [52]. MOO problems contain multiple conflicting objective functions and the solutions to such problems are mathematically speaking equally desirable. Hence, solving these problems requires a decision maker (DM), a domain expert, who can distinguish the most preferred solution [52]. In the MOO literature, the focus has generally been on having a DM or group of unanimous DMs, as in [52, 79].

However, real-world problems often involve more than one DM. When there are multiple DMs considering a decision problem, it is referred to as group decision making (GDM) [38, 49, 52, 67, 79]. GDM problems include a plethora of different situations and have existed among people as long as there have been groups of people and decisions to make. Naturally, in a wide problem area such as GDM, many different multi-disciplinary research fields ranging from political sciences [36] to economics, behavioral sciences and psychology [68, 82] and to mathematics and computer science have emerged [81].

Depending on the type of a GDM problem under consideration, different decision rules are followed to reach an agreement among the group [37, 38, 43, 67]. For example, the parliament's decision making process relies on voting rules. Or a group of colleagues may try to find a consensus, and disputants trying to settle a lawsuit rely on a lawyer to facilitate an agreement. There are advantages and disadvantages to solve problems using GDM. Some of the advantages include that having more people involved in the decision making gives more expertise in total to solve the problem, and the participation and act of working as a group will lead to a better acceptance of the decision by the group [67]. The disadvantages include properties such as coordination loss, overload in communications and in cognitive side and interpersonal conflicts [67]. Coordination loss means that some of the efforts of the individuals will be spent to coordinate the actions of the group.

Let us imagine two examples of very different types of GDM problems. The examples are designed to highlight the vast difference of possible GDM problems regarding the participants and their relations to each other. A simple case, familiar to everyday life, would be a family (on good terms with each other) choosing a restaurant. The family members, the DMs, may have different preferences regarding what the restaurant should offer. For example, one DM may prefer an Italian restaurant, another one may want a restaurant with vegetarian options. The DMs may discuss, negotiate or vote about the choice. They may argue, but after all, they are going together to the selected restaurant. This is an example of a collaborative GDM problem, where the DMs acknowledge the existence of a common problem, and attempt to work together in a friendly manner trying to reach a group decision [49]. The decision rule is based on the consensus among the participants.

A very different GDM problem would be the following. Imagine the days of ancient Rome, siege negotiations between the Roman general outside the walls and the petty king of the city under siege. The DMs are enemies, hence heavily conflicting, and there are thousands of lives at stake. The DMs' preferences are also heavily conflicting, and their aims are opposite. The general wants to take over the city and the petty king wants to stay as an independent ruler. The general has to worry about the consequences of a prolonged siege, and maybe deadly and possibly failing assault. The petty king knows that if the negotiations fail, and they fail to defend the city, the Romans will not show mercy and the city will be pillaged. Both sides must decide on what terms they would settle to end the siege or, whether it is more beneficial to end the negotiations. For example, the Romans may wish that the city gives up unconditionally and becomes a subject of Rome. However, the petty king may wish that with successful negotiations, a pledge of support and a moderate amount of gold steers the Romans away. This is an example of a non-cooperative GDM problem [49], where the DMs act as adversaries or disputants. Here, the decision rule is based on the negotiations and there is a threat of consequences, in case the negotiations end unsuccessfully. In these types of GDM problems large conflicts, competition and returning to the status quo - not making any decision - are common occurrences [67].

In the systematic literature review presented in this thesis, we focus on GDM, where there are multiple different human DMs involved, each with their own opinions, attitudes and preferences regarding the decision problem. The DMs recognize the existence of a common problem and attempt to reach a collective decision or a group decision (the terminology used in the literature varies) [49]. In the context of MCDM, GDM can be classified into three groups: social choice theory, game theory, and expert judgment/participatory group [33, 37, 43]. In social choice theory, each DM casts a vote or several votes depending on the method to select the favorite solution. The counting process of the voting relies on social choice or a welfare function and will provide the group decision. Game theory is concerned with situations of conflicts of interest, where individuals are pursuing their interests against other players who are pursuing their interests. The decision problem to solve takes a game form, where the players deploy different strategies to try to win. Game theory provides different automated strategies and bargaining processes for decision making [67]. The expert judgment considers using DM's preferences and suggestions of solutions, possibly considering new solutions, and the selection of a solution is based on agreement among the group [37].

Much of the GDM in the MCDM context has focused on problems, where there is an explicitly given set of alternative solutions and the problem is solved by aggregating the preferences of the DMs of the alternative solutions [43, 49]. The aggregation provides a collective alternative solution as the final solution. There is a lot of literature in the GDM considering problems with explicitly given sets of alternatives, we refer to e.g. [16].

However, as mentioned earlier, in MOO, solutions are not explicitly known beforehand. In this case, the approach of aggregating the preferences may lead to a collective solution that does not exist. At the same time, there may be an infinite number of solutions for DMs to give preferences. This is the main difference that makes much of the existing literature on GDM poorly applicable to MOO with multiple DMs. In this literature review, we are interested in the problems that do not have an explicitly given set of alternatives. We aim to understand how the multiple decision makers interact with the MOO problem, and how the group converges on a single solution to implement.

As far as the rest of this thesis is concerned, in Chapter 2, key concepts of MOO are introduced. This chapter discusses what the DM is and preferences and the main methods used in MOO. Chapter 3 introduces the combination of GDM and MOO and discusses relevant findings from the literature. In Chapter 4, we present the procedure of the systematic literature review. We present the results of the review in Sections 4.1, 4.2 and 4.3. In Chapter 5, we discuss the results of the literature review, the limitations of the study and possible future research topics. Finally, we provide conclusions in Chapter 6.

2 Multiobjective optimization

In this chapter, an introduction to the concepts of multiobjective optimization is given. First, in Section 2.1 the formulation of a multiobjective optimization problem is presented with relevant key concepts. In Section 2.2 we define what a DM in this thesis is and how the preferences given by the DM can be used to solve multiobjective optimization problems in different ways. Then, in Section 2.3, we briefly discuss some well-known multiobjective optimization methods.

2.1 Problem definition and important concepts

A multiobjective optimization problem (MOP) [52] can be defined as

$$\text{minimize} \quad \{f_1(x), f_2(x), \ldots, f_k(x)\}, \text{subject to} \quad x \in S \subset \mathbb{R}^n, \tag{2.1}$$

where k ($k \geq 2$) denotes the number of *objective functions* $f_i : S \rightarrow \mathbb{R}$. The vector of objective function values can be defined as $f(x) = (f_1(x), f_2(x), \ldots, f_k(x))^T$. The decision variable vectors, called *decision vectors*, $x = (x_1, x_2, \ldots, x_n)^T$ belong to the feasible set S (defined by constraints), which is a subset of the decision variable space $\mathbb{R}^n$. The image of the feasible set S, is denoted by $Z(= f(S))$, and it is called a feasible objective set. The members of Z are called *objective (function) vectors* and are denoted by $f(x)$. The words in parentheses are often omitted for short. Each of the objective functions in a MOP can be either minimized or maximized and transforming a minimization problem into a maximization problem can be simply done by multiplying the objective function by -1 and vice versa. In what follows, we assume minimization unless otherwise mentioned. When a MOP consists only of two objective functions, it is called a biobjective optimization problem [44].

In multiobjective optimization problems (MOPs) it is assumed that there exists at least partial conflict between the objective functions, which means that the problems do not have a single global optimum [52]. Instead, there is a set of Pareto optimal solutions. A solution is Pareto optimal if no values of any objective functions can be improved without declining some other objective function values. In other terms, a solution $x \in S$ is Pareto optimal, if there exists no other solution $x^* \in S$ for which $f_i(x^*) \leq f_i(x)$ for all indices i and the inequality is strict for at least one index [52]. The image of a Pareto optimal solution is called a Pareto optimal objective vector. The set of Pareto optimal objective vectors is sometimes called a Pareto front. A Pareto optimal set, called Pareto set for short, is the set of Pareto optimal solutions. There can be an infinite number of Pareto optimal solutions in the Pareto set.

A vector representing minimum values of the objective functions is called an ideal (objective) vector [79]. The ideal vector is infeasible, otherwise, it would be the only solution in the Pareto optimal set. The ideal vector provides the lower bounds in the Pareto optimal set for each objective function and it is denoted as z^*.

The term dominate is useful when considering different objective vectors. Let us consider two objective vectors in Z. An objective vector $f(x_1)$, dominates 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 [83]. An objective vector is called non-dominated if it is not dominated by any other feasible objective vector [83].

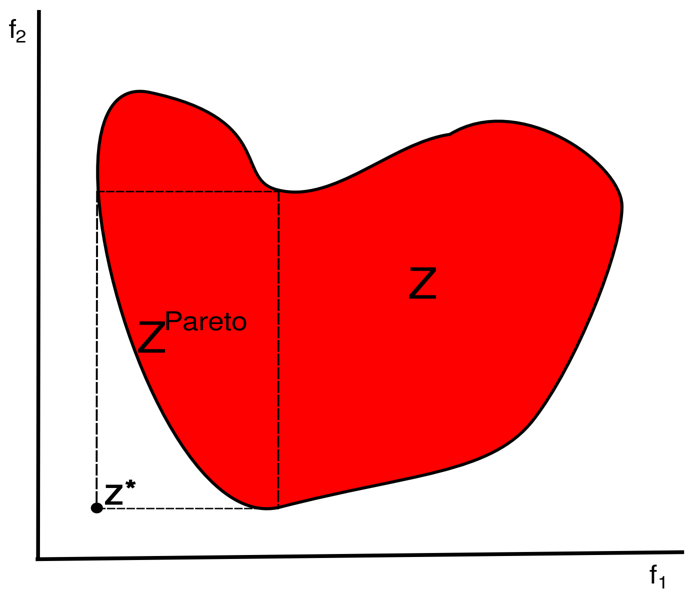

Figure 1. Relevant concepts in MOO shown graphically for a biobjective optimization problem. The picture contains the image of the feasible objective set Z (the whole red area), the Pareto set Z^{Pareto} (the line segment inside the dashed line box) and the ideal vector z^*.

Some concepts of MOO that have been presented in this section are visualized in Figure 1 for a biobjective optimization problem. The concepts apply with more than two objective functions. Here, a biobjective optimization problem is shown to ease visualization.

Finding a solution to a MOP in one way or another is called a solution process [52]. Various MOO methods can be used to solve MOPs. In the absence of any new information since all of the Pareto optimal solutions can be regarded as equally good in a mathematical sense, there is a demand to find as many Pareto optimal solutions as possible [24]. This type of search for the representation of the Pareto optimal solutions is the main focus e.g. in a majority of the evolutionary multiobjective optimization (EMO) literature [20].

However, when there is a DM present, the subjective preferences of the DM can be used to determine a most preferred solution [52]. Such a solution is referred to as the final solution. Hence, the goal of solving the MOP is to select the final solution. In this review, the MOP is not solved until the final solution has been selected. In the next Section 2.2, we discuss what is the DM and how the DM can affect the solution process of the MOP.

2.2 The decision maker and preferences

In the context of this review, the DM is a human who is an expert in the problem domain. In this section the DM is either a single person or an unanimous group of persons. Later in this review, we discuss multiple DMs. This person may be someone having some stake in the resolution of the solution process, called a stakeholder. Either way, they are referred to as DMs. In some research domains, a DM may be an artificial agent DM, either representing the preferences of a DM or acting upon decision rules set upon it. In this review, we consider only the former type of agent DMs.

There are some assumptions regarding the DM in this review. Firstly, it is assumed that the DM can provide preference information and eventually distinguish the final solution to the problem. The preference information can be expressed in various ways and it is used in the solution process. Secondly, we assume that the DM always prefers Pareto optimal solutions over dominated solutions. In other words, in case of minimizing all objectives, the DM always prefers less to more. Therefore, the final solution is a Pareto optimal solution, which the DM deems as the most preferred solution.

The role of the DM is to give preference information that can be used in solving the MOP [52]. If preference information is available, it can be incorporated into the solution process before, after or during the optimization. The following classification of MOO methods relies on the participation of the DM in the solution process and the timing when the DM expresses their preference information [52]:

1. No preference: No articulation of preference information is used and some neutral compromise solution is selected.
2. A priori: The DM provides their preferences before optimization. The optimization process tries to find a Pareto optimal solution satisfying the preferences as well as possible.
3. A posteriori: A representation of the Pareto optimal solutions is first found. The DM provides their preferences by selecting the desired solution after seeing the solutions.
4. Interactive: The DM iteratively articulates their preferences during the optimization process and can update the preferences.

The advantages and disadvantages of a priori methods include, that it is often difficult for the DM to know in advance what type of solution(s) they would prefer [86] and preferences may be optimistic or pessimistic. This may lead to the optimization process of generating non-relevant solutions. However, a priori methods do not require much of the DM's time to participate in the solution process.

In a posteriori methods, the advantage is that the DM has a lot of good solutions to choose from, after a representative set of Pareto optimal solutions has been found. This information may lead to a better understanding of the trade-offs between the objectives [86]. However, the disadvantages include that the computation of the representation may waste time and resources on solutions the DM is not interested in. In addition, the DM may have difficulty evaluating possibly dozens or hundreds of solutions, so choosing a subset of solutions to show to the DM is another issue to solve.

Interactive methods contain iterations consisting of phases of preference elicitation and solution generation. The phases take turns until the DM finds the final solution (or some other stopping criterion is met) [55]. The DM expresses their preference information during the preference elicitation steps. The advantages of interactive methods include involving the DMs in the optimization process, which can help the DM to learn about the problem and their preferences [53]. This provides other advantages such as by having more understanding of the problem, the DM can better justify why they prefer the final solution over the other solutions [20, 55]. However, the disadvantage of this is that an interactive solution process requires much more time and focus from the DM [55].

Preference information types

In the following, we introduce relevant preference information types needed in this literature review. Some of the preference information types we introduce are widely used in MOO and some are used more often in MCDA. However, the aim of the following discussion is not be exhaustive or try to give a wide overview of the different types of preference information in MOO: we aim merely to introduce all the preference information types that are used later in this review.

The DM may express preference information in various ways and the preference information expressed may be related to objective functions or solutions [20]. The preference information may consist of weights for or classification of objective functions, local preference information such as aspiration levels, the DM indicating the most preferred (or disliked) solution(s) of a subset or pairwise comparisons of objective functions or solutions [53, 74, 86].

Weights for objective functions are supposed to represent the relative degree of importance of the objective function for a DM, often modeled as a weight vector with components $w_i \geq 0$ for all $i = 1, \ldots, k$ for k objective functions. However, as shown in [70], it is not clear what this relative degree of importance of the objective functions actually means for a DM as the weights behave in an indirect way [55]. For example, small adjustments of the weight vector may cause big differences in the objective function values and vice versa [55]. Moreover, as the weight values are in the weight space [86], they do not have a clear meaning for the DM who is a domain expert and understands the objective space [52, 83]. In addition, it is very difficult for the DM to adjust the weight vector to obtain a desirable solution since there does not exist any correlation among the weight value w_i and the objective function value $f(x)$ corresponding to the following solution x [59]. Hence, it can be hard for DM to provide an accurate weight vector or get desired solutions by modifying the weights [59, 83, 86]. In Section 2.3, we mention some other issues of using weights for objective functions in solving MOPs.

Reference points are formed of aspiration levels, which are desired values given by the DM for each objective function or reservation levels, which are values that the DM expects to be achieved [52, 74, 88]. The aspiration (or reservation) levels form the vector referred to as a reference point and they have some useful properties, such as, the reference point does not require any consistency in the preferences the DM expresses among iterations and the reference point reflects the desires of the DM well [59]. Moreover, the reference point is in the objective space [86] and since the objective functions are meaningful to the DM, the reference point can be argued to be more understandable for the DM than weights for objectives [59].

The DM can rank solutions in the order of satisfaction, referred to as the ranking of solutions [20]. The ranking of solutions requires some way of determining the subset of solutions that is shown to the DM as in most cases there are more solutions than a human DM can be expected to rank. Alternatively, the DM can only indicate the most preferred solution(s), which most satisfies the DM. Similarly, the DM may indicate a least preferred solution(s) from the shown solutions.

In pairwise comparisons, the DM is shown two solutions and the DM indicates the preferred solution of the two [13]. Often the solutions shown in pairwise comparisons are selected randomly, although they can be selected more carefully e.g. according to the earlier preference information given by the DM [13].

The preferences expressed by a DM can be transformed to a preference model to evaluate the decision process of the DM [79]. A preference model that is commonly used is a utility function [43]. This assumes that the DM makes decisions based on a utility function $U : R^k \to R$, representing the preferences of the DM of the objective functions [43]. By using the utility function, the preferences of the DM of the different solutions can be determined. The utility function can be inferred from some of the preference types given by the DM e.g.

from a ranking of solutions [20].

One more preference information type we define in this review, is called selected objectives. This concept means a subset of all the objective functions in the MOP that a DM cares about.

Preference relations relate to the situation that for a DM all the objective functions or solutions may not be equally important. In this way, the objective functions can be ranked in a preferred order, called as ranking of the objectives. Preference relations are binary relations, that have properties such as transitivity [25, 79]. Transitivity means that if we have solutions x, y, z, and the following is true, x is preferred to y and y is preferred to z, then also x is preferred to z [79]. The main disadvantage of preference relations is that they cannot handle non-transitive preferences [20].

Outranking is a different ranking of the solutions allowing non-transitivity [86] and it is often used in MCDA [71]. The outranking-based methods discussed in this review are based on building outranking relations between the solutions. In outranking methods, the aim is to determine for each pair of solutions whether the DM prefers one of them, is indifferent or the solutions are incomparable [20]. To achieve this, different preference indicators are defined and compared with specific threshold values (which the DM may give as preference information). The disadvantages of this approach are that the DM is required to give many different parameters [86] and the method may become computationally expensive, especially when there is a large number of solutions [20].

Uncertainty in the preferences

The DM may prefer to articulate their preferences with uncertainty [86]. In contrast to earlier *precise* preference information types, uncertain preference information is referred to as *imprecise* preference information. In cases like this, to model the uncertainty in preferences, small deviations can be introduced into the preferences. This can be modeled by the DM providing the preference information as interval or fuzzy numbers. An interval number is an extension of the concept of a real number, described as a range $E = [l, u]$, where l is the lower limit and u is the upper limit. The range contains all real numbers lying between the lower and upper limit [29]. Interval numbers can be used in interval preference models, e.g. the interval outranking model in [27].

Fuzzy numbers can be used to handle uncertainty in preferences such as reference points, weights and outranking [86]. For example, considering a reference point, denoted by $R = (r_1, \ldots, r_k)$ for k objective functions using fuzzy numbers, the desired value of each objective can be expressed as triangular fuzzy numbers $r_i = (r_i^{lower}, r_i^{most}, r_i^{upper})$, where r_i^{most} is the most possible value of the fuzzy number and r_i^{lower} and r_i^{upper} are, respectively, the lower and upper bounds, according to the fuzzy preferences of the DM [89]. A membership function, a generalization of an indicator function, can be used to represent the degree of truth of the fuzzy reference point [57]. Used in MCDA problem settings, in fuzzy preference relations, the DM's preferences are described with a membership function that denotes the preference intensity of a solution over the other alternative solutions [35]. There are different types of fuzzy preference relations in the literature such as additive, multiplicative or linguistic preference relations [35, 45].

In this review, the preference information type that the DM expresses has the main focus, not whether the preference information is given in a precise or an imprecise format. Now, we have discussed the role of the DM in the solution process and what type of preferences the DM may express. Next, we discuss a few MOO methods used to solve MOPs.

2.3 Introducing two methods for solving MOPs

In this section, we discuss two MOO methods that are used or inspired the developed methods in 70% of the papers in the reviewed GDM literature. They are introduced and some of their main advantages and disadvantages are discussed due to their overwhelming popularity in the literature. The discussion here will introduce concepts that are later referred to. In Chapter 3, we consider group decision making in the context of MOO.

One way to classify MOO methods is to split them into scalarization-based and evolutionary-based methods. A scalarization-based method refers to solving a MOP by scalarization. Scalarization means that the MOP is converted to a single objective (scalar-valued) optimization problem [52]. The main advantage of scalarization is that when the scalarization method is selected properly (considering the specifics of the problem to solve), the Pareto optimality of a found solution can be guaranteed [52, 53, 79]. There are various scalarization-based methods.

The advantages of scalarization-based methods include that they are easily adapted to different types of MOPs and DM's preferences and they are computationally efficient [52]. The disadvantages contain that to find multiple Pareto optimal solutions the scalarization methods often have to be applied multiple times [24] and the solution found is significantly impacted by the choice of the scalarization function [54].

An example of a scalarization method is the weighted sum method, presented, e.g. in [83, 90]. The idea of the weighted sum method is to connect each objective function with a weighting coefficient and minimize the weighted sum of the objectives. In this way, the multiple objective functions are transformed into a single objective function [52]. It is typically assumed that the weights $w_i \geq 0$ for all $i = 1, \ldots, k$ are normalized so the following applies, $\sum_{i=1}^{k} w_i = 1$ and the objective functions should be normalized [55]. A solution to the weighted sum method is Pareto optimal if the weights are positive or the solution is unique [52].

As mentioned, the weighted sum is used a lot in the reviewed literature. The advantages of the weighted sum method are that it is simple to understand and implement, and it can be used in an a priori or in an a posteriori manner [52] and also in an interactive manner when the DM can adjust the weights [59, 83]. However, the weighted sum method contains several disadvantages. Firstly, the weighted sum method cannot find unsupported solutions of the Pareto front in nonconvex problems [59]. Secondly, the correlations among the objective functions may lead the DM not being able to find desirable solutions as a "good" weight vector (according to the preferences of the DM) may provide an undesirable solution and vice versa [83]. Lastly, the weighted sum method is likely to find solutions with unbalanced objective function values [55] and an even distribution of weights may not result in an evenly distributed set of solutions [44].

Evolutionary methods are heuristics for solving MOPs and there is a plethora of different EMO methods, called evolutionary multiobjective optimization algorithms (EMOAs) [4, 23]. EMOAs model evolutionary processes occurring in nature by creating a population of individual solutions that are evaluated for their fitness [4]. The fitness measures the goodness of a solution, and how well the solution performs on the problem. Naturally, in the MOO context, the better the fitness of the solutions is, the closer it approximates a Pareto optimal solution [23]. As EMOAs are methods based on heuristics, they cannot guarantee Pareto optimality [22] and hence we refer to these solutions as approximated solutions. However, the EMOAs can guarantee that the solutions found at the end of the method are non-dominated solutions [22], meaning that no solution belonging to the final population dominates any other solution in the final population.

The evolutionary operators such as crossover, mutation and selection [4] modify the population. Briefly, the selection operators select (in some specified manner) some individuals for reproduction. Reproduction contains crossover and mutation operators. Crossover combines the selected individuals in some manner, and the offspring are subjected to random mutations adjusting their properties. The process continues over multiple generations with the fittest individuals surviving and individuals with a low fitness being removed from the population.

EMOAs have some advantages and disadvantages. The advantages include i) the EMOAs are less likely to get stuck on local optimums, ii) EMOAs are simple to implement, and iii) EMOAs are flexible to use and can be applied in a wide range of MOPs including nonconvex objective functions [21, 23, 24]. The disadvantages contain that EMOAs cannot guarantee the Pareto optimality of the found solutions [23] and EMOAs require many function evaluations, which is an issue, especially with computationally expensive objective functions [39]. In addition, EMOAs have several parameters (such as population size, crossover, mutation and selection operators to use and their probabilities) that need to be selected in some manner and they affect the performance of the methods [39].

An example of an EMOA is the non-dominated sorting genetic algorithm II (NSGA-II) [24]. NSGA-II is an a posteriori method, which aims to generate a diverse set of non-dominated solutions. To get started, NSGA-II generates an initial population and evaluates the fitness of the solutions in the population. A non-dominated sorting procedure is utilized to sort the individual solutions into different levels, called fronts, based on the dominance relationships with respect to each other. Then, a crowding distance operator is used to measure the distances between the solutions in the front to maintain diversity among the population. NSGA-II can use, for example, binary tournament selection as the selection operator and simulated binary crossover operator and polynomial mutation to create a new population from the selected solutions. The selection and reproduction operators generate the new population, and the process is repeated for a specified number of generations.

3 Group decision making

In this chapter, we introduce the combination of group decision making and multiobjective optimization. As discussed in Chapter 1, the main difference in MOO with multiple DMs to most GDM problems in MCDA is in the properties of the feasible solutions. The solutions are not explicitly given in MOO, instead, they are found using MOO methods. First, in Section 3.1, we introduce the combination of GDM and MOP and define relevant concepts regarding the group of DMs and how they can collectively make decisions. In Section 3.2, we discuss how the multiple preferences (from several DMs) can be incorporated into the solution process. Lastly, in Section 3.3, we briefly consider relevant concepts from the GDM literature used in determining the group's preferred solution.

3.1 Introducing GDM-MOP

The combination of GDM and MOP can be referred to as a GDM-MOP. The term, GDM-MOP, was defined first time in [89] and after that, the term has been used in [6, 27, 29, 57, 76]. However, in [89], the authors state that the goal of solving a GDM-MOP is to select the most acceptable solutions according to the group among the Pareto set. In this review, we assume that in solving GDM-MOPs, the group's aim is to select a most preferred solution (in the context of the group), called a collective solution. When multiple solutions are found using the multiple preferences of the DMs, they are referred to as collective solutions. The collective solution that is selected to be implemented in practice is called the final (collective) solution. The word in parentheses is usually left out.

The GDM-MOP is defined as a decision situation including:

- A group of m DMs ($m > 1$).
- A common MOP under consideration as defined in Equation (2.1).
- The goal is to find a collective solution acceptable to the group.

The following aspects regarding multiple DMs are important and, from now on, they are referred to determine a *group structure*. The DMs have their own knowledge, attitudes and opinions, and the DMs recognize the existence of a common problem and attempt to reach a collective solution [43, 49]. The DMs in the group may have different roles or degrees of importance in making the decision. For example, the group may be led by a supra DM (SDM) [43], who has the final say over the final solution. The group may or may not be able to communicate in-person. The group may prefer making decisions anonymously or in-person [51]. The group may prefer making the decision by some specified way, e.g. by group discussion or by a specified decision rule e.g. by voting. In this review, we refer to this specific way or rule as a *group decision method*. A group decision method is some sort of a principle or criterion that provides the final solution. Commonly used group decision methods include group discussion, negotiation, aiming for consensus, voting and relying on a third party to resolve conflicts between the DMs [49].

Finding a solution to a GDM-MOP in one way or another is called a solution process. We define group decision making in multiobjective optimization (GDM-MOO) methods to refer to the methods used to solve GDM-MOPs. The concepts of MOO described in Section 2.2 apply to GDM-MOO methods. In addition, the DMs can express their preferences using the same types of preference information and the preference information of the DMs can be incorporated a priori, a posteriori or in an interactive manner. Hence, the GDM-MOO methods can be classified into a priori, a posteriori and interactive GDM-MOO methods [57].

In Section 3.2, we discuss the approaches found from the reviewed literature of handling multiple preferences (from several DMs) in the GDM-MOO methods. We also study how these approaches can be combined with group decision methods to find a final solution that is acceptable to the group.

3.2 Incorporating multiple preferences

In this section, we consider approaches to handling multiple preferences (from several DMs). The multiple preferences must be somehow incorporated into the GDM-MOO method to find collective solutions in the solution process. In this thesis, we distinguish between two types of handling preferences and refer to them as an indirect and direct approach. When referring to them together, they are called preference approaches.

Indirect approach

In an indirect approach, the individual DM's preferences are in some way combined into a collective preference. This can be done in different ways, however, often different aggregations operators are used. An example of an aggregation operator is the arithmetic mean. The aggregation of the preferences is crucial and different aggregation operators lead to different collective preferences. The SDM may also be in charge of forming the collective preference. The collective preference can be formed using any preference information type, that can be mapped from multiple preferences into a single preference. For example, multiple reference points can be aggregated into a collective reference point or multiple weight vectors into a collective weight vector. Similarly, one can aggregate multiple utility functions into a collective utility function. They all can be referred to as collective preferences.

The collective preference can be used similarly as a single preference in various preference-based MOO methods for a single DM. Depending on the MOO method applied, either a collective solution or multiple solutions will be reached.

The issues of the indirect approach have received a lot of research in GDM [35, 40]. By purely aggregating the preferences of the DMs into a collective preference, the existence of an agreement among the DMs cannot be guaranteed, which may lead to a solution for which some DMs feel that their preferences have not been considered [45, 69]. In the case of the DMs' preferences being about shown solutions, the aggregated collective preference may not correspond to any solution in the Pareto set. For example, an average of two solutions may not exist. Another issue to mention is that, even if the individual preferences (e.g. complete rankings of the solutions) are transitive, the collective preference is not, as discussed in the famous Arrow's Impossibility Theorem [3].

Direct approach

In a direct approach, the multiple preferences from several DMs are not combined into a collective preference. Instead, the preferences are used *directly* in some way in the solution process. For example, the method may use multiple weight vectors from multiple DMs to find solutions according to each of the weight vectors as in [47]. Another example is in [64], where the utilized EMOA prefers solutions that are closer to reference points of individual DMs. The direct approach does not exclude the possibility of using also e.g. a collective reference point along the individual preferences as in [89]. The main difference to an indirect approach is that there are multiple individual preferences instead of a single collective preference.

However, the issue with the direct approach is that an another step need to be taken to select the final solution. After all, in solving GDM-MOPs, a final solution must be determined. Hence, another decision is to be made, where the DMs must choose the group decision method to use to select the final solution.

Finding the final solution

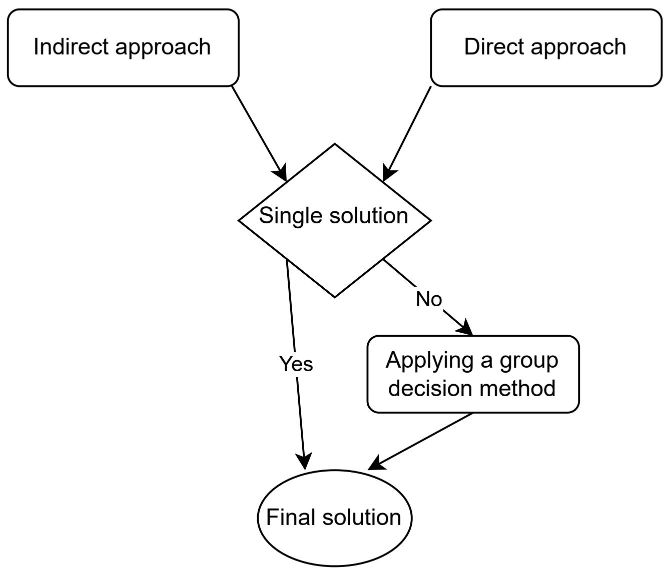

Figure 2. A graph illustrating how indirect and direct approaches can be used to find the final solution.

Figure 2 illustrates how indirect and direct approaches can be used to find the final solution. As mentioned, in the indirect approach, depending on the applied MOO method either a collective solution or multiple solutions are found. In the case of a collective solution, that solution can be defined to be the final solution. For example in [11] the collective preference was used in a weighted sum method to find a collective solution, determined as the final solution. In the case of multiple collective solutions, another group decision method is needed to select the final solution.

A direct approach requires utilizing a group decision method at some phase of the solution process to converge from multiple solutions (generated using multiple preferences) into a final solution. A group decision method can be incorporated into the GDM-MOO method to determine the final solution. For example, the DMs may vote about the solutions, and the solution with the most votes is selected as the final solution.

3.3 Introducing relevant concepts found in the literature

In this section, we introduce relevant concepts found in the GDM literature. First, we discuss using consensus as a group decision method and as a way of tackling the issue of aggregating multiple preferences into a collective preference. Then, we introduce group decision methods, which are referred to later in this review.

Introducing consensus in GDM

As mentioned in Section 3.2, the issues with an indirect approach are that the agreement of the final solution cannot be guaranteed and that the collective preference may not directly relate to a Pareto optimal solution. Moreover, in problems with multiple DMs with heterogeneous preferences and multiple conflicting objectives, it seems unreasonable to expect the group to be able to attain full unanimity. Since the acceptance of the final solution is crucial in many real-world GDM problems, a phase called a *consensus reaching process* (CRP) can be added to the solution process [45, 63].

Next, we discuss what a consensus in the context of GDM is and how consensus can be defined. There are several different consensus definitions in the literature, and the term has been used for years in various contexts [34]. As mentioned, in MCDA problem settings often used in GDM, the preferences are related to the solutions and in MOO the preferences can be related to the objective functions or the solutions. Here, we discuss consensus in general, including consensus about the preferences and consensus about the solutions. Consensus can refer to a full agreement [34, 45], even though more flexible interpretations of a consensus have also been used considering a partial agreement [40]. We refer to [34] for different consensus definitions and ways of measuring consensus.

In this review, we define a *consensus* to mean an agreement among (most of) the DMs. A *consensus degree* refers to the level of agreement among the DMs. The consensus degree can be computed using a *consensus measure*. The consensus measure indicates how close the DMs' opinions are to unanimity [63]. Consensus measures can be classified as based on distances from a collective preference and based on distances between the preferences of the individual DMs [45, 63]. These distances are then aggregated using certain aggregation operators [91]. A consensus measure can be used to evaluate the consensus degree of a given solution. However, the acceptable consensus degree must be decided by some threshold. A *consensus threshold* indicates the minimum value of acceptable agreement [15, 45]. For example, at least 9 out of 10 DMs must agree on a given solution. We refer to [15] for more discussion on how to determine consensus thresholds. Hence, the solutions that satisfy the consensus degree are referred to as *consensus solutions*.

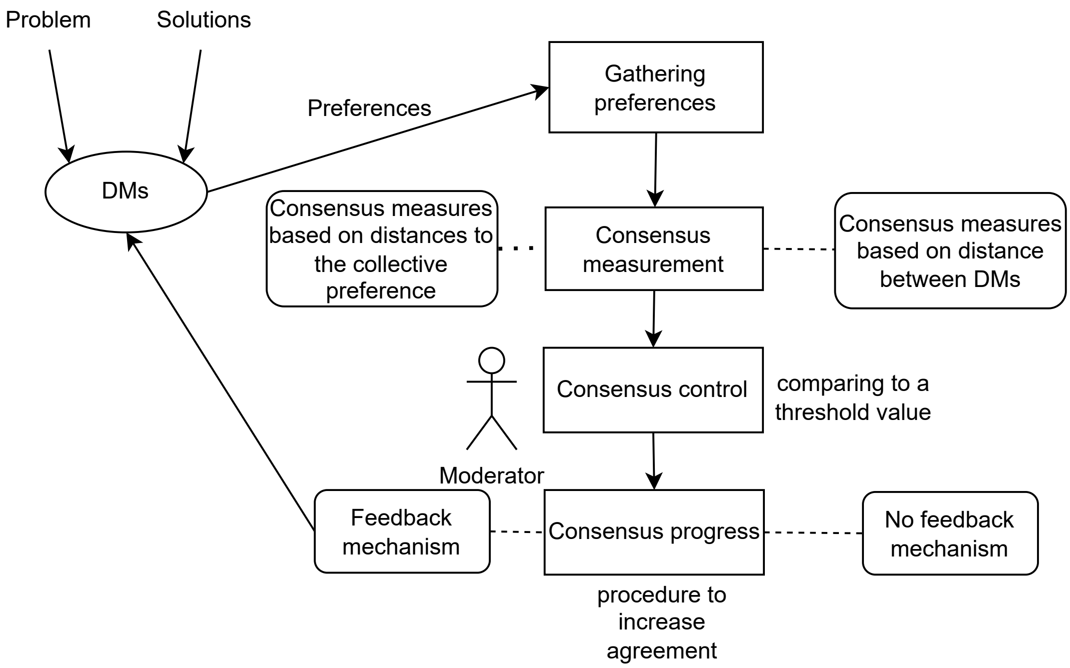

Figure 3. A general CRP scheme introduced in [45].

A consensus reaching process is an interactive process consisting of several rounds of discussion and negotiation to achieve a maximum degree of consensus among DMs [6, 14, 27, 34, 45]. The consensus reaching process involves a moderator (or an SDM who acts as the moderator) who knows the consensus degree of the group and is in charge of guiding the process in each round of the consensus reaching process by computation of some consensus measures [34]. The consensus reaching process can be performed in person or online.

Figure 3 has a general consensus reaching process scheme, adapted from [45]. In the consensus reaching process, first, the preferences of the DMs are obtained, and then the consensus degree can be measured. The consensus control stage refers to comparing the current consensus degree to the consensus threshold. The consensus progress stage refers to different ways of adjusting DMs' preferences. In consensus reaching processes with a feedback mechanism, the moderator provides feedback to the DMs to modify their preferences. In contrast, in no-feedback methods, the moderator adjusts the preferences of the DMs to be more like the collective preference. The consensus reaching process can successfully end when the consensus threshold is reached. The consensus reaching process continues with the DMs adapting their preferences if the consensus threshold is not met.

For a more detailed discussion and taxonomy of consensus reaching processes, we refer to [14, 63]. The consensus reaching process can be used in different stages of solving the GDM-MOP [6]. For example, in an a priori method, the consensus reaching process can be used to make the DMs' preferences more similar, in an a posteriori method to help to select the final solution and in an interactive method during the iterations to adjust the DMs' preferences.

Group decision methods

In the following, we discuss different group decision methods used in the reviewed literature. The group decision methods discussed here assume that we have generated a subset of solutions that are shown to the DMs. More specifically we have a finite number of j alternative solutions. In other words, we have an MCDA problem setting.

The group decision methods of group discussion and negotiation refer to the DMs discussing or negotiating in person of selecting the final solution. In this review, we state it separately if these methods are not performed in person, and instead e.g. through an online negotiation platform. As mentioned, the consensus reaching process can be utilized as a group decision method.

Many well-known MCDA methods can be utilized in GDM problems when multiple preferences are aggregated into a collective preference. For example, in the analytic hierarchy process (AHP) [75], the DMs perform pairwise comparisons of solutions. Then, the AHP method computes relative weights for the solutions. The final solution is found by aggregating the relative weights (from multiple DMs) by using e.g. a geometric mean operator.

Next, we consider group decision methods related to social choice methods and the ranking of solutions. There are two types of approaches in voting-based group decision methods. The DMs' preferences can be used directly with a social choice function to select a single solution from the set of solutions or indirectly with a social welfare function which aggregates the individual DM's rankings of the solutions into a single social ranking [8]. Both types of methods are referred to as social choice (SC) methods in this review. Generally, the social choice methods aim to satisfy all DMs' preferences, meaning that in most cases the solution selected may not be a solution each DM would have individually preferred the most [1]. This type of majority influence is seen as desirable behavior in the context of democratically oriented decision making [8].

Let us next discuss examples of social choice functions. In majority voting, also known as plurality rule, the DMs vote and the solution with the highest number of votes is selected as the final solution [1]. Another social choice method is score voting. In score voting, the DMs vote for each solution with a scale e.g. (0-9 Likert scale) and the solution with the highest score is selected as the final solution [78].

Let us then give examples of social welfare functions. The Borda count method is a ranked voting method where the DMs rank the solutions in the order of preference [8, 30]. The solution the DM least prefers gets a score of zero, the following solution gets a score of 1 and so on, up to j points for the most preferred solution. Then, the Borda ranking is decided by ordering the Borda scores. It is well known, that in the Borda count method, a determined minority group of DMs can lower the chances of otherwise popular solutions [8].

The Borda count is a linear function of a solution's average rank. Hence, the Borda ranking is the same as the ranking that would be achieved by taking an average of the preferences of the solutions [8], which may not be desirable in some situations. The following social choice method takes this issue into account. In a social choice method called a median voting rule, the DMs rank the solutions as above. But instead of averaging the ranks, the median of the ranks is taken. In this way, the solution that receives the highest score from the DMs is selected [1, 8]. Many different social choice methods can be utilized in selecting the final solution, and we refer to [8] for more discussion of social choice based methods.

The fallback bargaining (FB) methods simulate the bargains done in negotiations. First, the DMs rank the solutions. Then, the DMs fall back, step by step, to less preferred solutions, starting with their most preferred solution, then their second most preferred solution and so on, until a solution is found that all of the DMs agree with [12]. The different fallback bargaining methods utilize different agreement rules. For example, in the case of the agreement rule being unanimity, the method is called unanimity fallback bargaining. In some fallback bargaining methods, the DMs may be allowed to indicate *impasse* in their ranking, determining the solution they will not descend below in their ranking.

4 Systematic literature review

In this thesis, we conducted a systematic literature review concentrating on published papers in GDM-MOPs. The main goal of this review was to find out what kind of research has been conducted on solving GDM-MOPs. At the beginning of the process, there was no clear path or terminology to which we could refer to. Hence, we had to identify methods and definitions commonly used in solving these types of problems and tie them all together to the existing MOO literature. This work brought up e.g. the combination of GDM and MOP to be called GDM-MOP.

First, we discuss the goal for the systematic literature review including the research questions. Secondly, we introduce the way of the systematic literature was conducted and lastly, we discuss the search queries and databases used in the review.

The systematic literature review aims to find some often-used ways of solving MOPs and provide sensible classifications that would help the future work on this interesting amalgamation of GDM and MOO. The literature review investigates the state-of-the-art in GDM-MOO. The following research questions have been adjusted with the knowledge found during the process.

The main research question: What kind of methods exist in the literature to solve GDM-MOPs? To tackle the main research question, we can split it up into the following sub-research questions.

1. What kind of preference information is asked from the DMs in GDM-MOO methods?
2. How are the multiple preferences from several DMs incorporated into the solution process?
3. How is the final solution selected in GDM-MOO methods?

The way of conducting the systematic literature review followed loosely the steps of [61] adapted to the field of MOO. The purpose of the literature review was to assess the current literature regarding GDM in MOPs. A set of test searches was performed first to find proper search terms for the search queries. The test searches were conducted to find knowledge about relevant keywords and the combinations of the keywords.

A paper was included in the review according to the inclusion criteria if the title, abstract or keywords contained any of the search terms or any new but relevant combination of them. In addition, a paper was included if it was written in English and published as a journal or conference article, and according to the abstract was relevant to the literature review.

A paper was excluded if it did not consider a MOP. Moreover, if the problem was explicitly given as a set of alternatives, the paper was excluded. A paper had to contain more than one human DM (or e.g. an agent representing a DM) and the DMs had to be involved in the solution process. In addition, couple of papers of clearly low quality were excluded of consideration (e.g. language issues).

Literature searches were conducted using Scopus and Web of Science as search databases, with the search queries listed in Table 2. Differences in the search queries between Web of Science and Scopus are related to the syntax of the search engine and they perform functionally in the same way. As mentioned, only papers published in English were considered and the main search was performed in July of 2022. The following steps of extracting the information from the papers, screening for inclusion and exclusion were performed iteratively multiple times. This process took a lot of consideration and several iterations had to be done during the process.

The set of included papers was supplemented with forward and backward searches from each paper. Forward search means that the papers that have cited the paper under consideration are reviewed with the inclusion criteria. Similarly, backward search means that we examine the papers that the current paper under consideration has cited. Again, inclusion and exclusion steps were performed. In the end, the total number of papers included was 40.

	Query 1	Query 2
Scopus	(TITLE-ABS-KEY ("group decision making) AND TITLE-ABS-KEY ("multi-objective optimization) AND NOT ALL ("multi-attribute))	(TITLE-ABS-KEY ("negotiation) AND TITLE-ABS-KEY ("multi-objective optimization))
Web of Science	"group decision making" (Topic) and "multi-objective optimization" (Topic) not "multi-attribute" (All Fields)	"negotiation" (Topic) AND "multi-objective optimization" (Topic)

Table 2. The search queries used in this literature review.

4.1 A priori methods

In this section, we discuss GDM-MOO methods, where preference information is given a priori and how it is incorporated. In Subsection 4.1.1, we answer the question about what kind of preference information is asked from the DMs. In Subsection 4.1.2, we discuss how the multiple preferences from individual DMs are incorporated into the solution process in an a priori manner. Lastly, in Subsection 4.1.3 we consider how a priori methods select the final solution.

In Table 3, the reference column indicates the paper referred to as on that row and the preference information types column mentions the utilized preference information types separated by a comma. In the cases where the DMs had options the word *or* is separating the options. The third column indicates whether the methods used indirect or direct approaches. The final solution column indicates whether the method identified the final solution for the GDM-MOP.

Reference	**Preference information types**	**Incorporating multiple preferences**	**Final solution**
[72]	reference point	direct	No
[50]	fuzzy ranking of objectives	indirect	Yes
OptMPNDS2 [80]	selected objectives	-	No

[11]	weights or ranking of objectives	direct	Yes
[73]	reference point	direct	No
[32]	-	indirect	Yes
OptMPNDS [48]	selected objectives	-	No
MWS-NSGA [47]	weights for objectives	direct	No
[31]	weights for objectives	indirect	No
[64]	reference point	direct	No
[89]	fuzzy reference point	direct	No
[58]	fuzzy reference point	direct	No
[57]	fuzzy reference point or triangular or Gaussian membership function	direct	No
WIN [17]	ranking of objectives	indirect	Yes
NSS-GPA [10]	reference point, deviation value	indirect	No
W-NSS-GPA [9]	reference point, deviation value	indirect	No
[28]	weights, fuzzy preference relations, reference point	indirect	Yes

Table 3: The reviewed a priori GDM-MOO methods in a table format.

4.1.1 Preferences in a priori methods

This subsection describes what type of preference information has been used in published a priori methods. As can be seen in Table 3, the most common type of preference information used in GDM-MOO methods is reference points. Reference points can be used with precise and imprecise preference information types. In the reviewed literature considering precise preference information types, the DMs articulate their preferences as reference points in [9, 10, 28, 72, 73]. Besides the reference point, in [9, 10], each DM provides information about how much they can tolerate deviation in each component of the reference point.

Sometimes the DMs' preferences may be regarded to be imprecise, in which case fuzzy, interval or triangular numbers can be used to model these imprecisions. In [28] the DMs set a fuzzy preference relation and in [58, 89] the DMs provide their preferences as triangular fuzzy numbers forming fuzzy reference points. In [57], the authors widen the possibilities of the types of preference information the DMs can express, as the DMs can define their preferences as a triangular or Gaussian membership function.

Other commonly used preference information types in the reviewed literature are DMs expressing the degree of importance of objectives as weights, as in [11, 28, 31, 47] and ranking of the objectives as in [11, 17]. In [50] the DMs linguistically express fuzzy ranking of the objectives. While the preferences are given by ranking the objectives in [11, 17, 50], they are transformed into weights of the objectives.

Finally, we mention other types of preference information that are not so popular. The preference information is incorporated in the problem formulation stage in [48, 80] where the DMs select the objectives they care about. No further involvement of the DMs is used in these methods. It is not clearly stated in [32] whether the DMs articulate their preferences in some unspecified manner or the preferences are in some way deducted from the DMs' known interest in the objectives. The following subsection focuses on handling the multiple preferences from multiple DMs.

4.1.2 Using multiple preferences in the solution process

In this subsection, the focus is on the approaches presented in the reviewed literature regarding the handling of multiple preferences. A priori given preferences can be used in various ways to help in finding collective solutions. As mentioned earlier, the ways can be roughly classified into using the preferences directly, or indirectly. In the cases where group interaction is possible, a consensus reaching process can be performed to bring their preferences closer to each other before running a priori method.

Indirect approaches

Agent-based negotiation support systems NSS-GPA and W-NSS-GPA presented in [9, 10] respectively, aim to aggregate the DMs' preferences that can be assumed to be at least partly conflicting before the optimization. As mentioned earlier, the DMs provide their preferences as reference points and also specify acceptable deviations for each component of the reference point. The DMs participate in several negotiation rounds using an agent-based software framework. An assistant agent represents each DM, and a moderator agent facilitates the procedure. The DMs communicate through the software framework making offers of changing their preferences and requesting others DMs to change their preference. The moderator agent regulates the negotiations based on the current consensus degree. The consensus degree corresponds to the average of all distances of the DM agents' reference points from the collective reference point. Based on the consensus degree, the moderator agent gives suggestions to the DMs to modify their preferences.

The main difference between NSS-GPA and W-NSS-GPA is the equality of the DMs and the way of handling the possible manipulative behavior of the DM agents. W-NSS-GPA considers different degrees of importance of the DMs, modeled by weights. In both methods, the moderator agent considers manipulations by the DM agents and penalizes manipulative DM agents. The penalization is done by changing their reference point in [10] and by manipulator isolation in [9]. The manipulator isolation deprives the manipulative DM agent from communicating with other DM agents and forces the manipulative DM agent to update their preferences according to the collective preference.

In [50], the DMs' preferences are gathered using a fuzzy ranking of the objectives. The DMs are regarded to be equal and by utilizing a multi-attribute decision making method, a fuzzy factor rating system is used to determine the objectives' collective weights, taking into account the DMs' preferences. Another approach utilizing fuzzy numbers is in [28], where the authors build a fuzzy outranking model from the weights for objectives, reference points, and fuzzy preference relations given by the DMs. The collective outranking model is used to determine the satisfaction of the DMs with the solutions generated using NSGA-II.

An approach utilizing negotiations to form collective preferences is applied in [31]. The MOP is decomposed into different subproblems. The DMs working on these subproblems negotiate about the parameters and weights for objectives used in scalarization methods such as the weighted sum method. The negotiated weights for objectives form the collective weight vector.

The MOP to be solved in [32] is related to a real-world problem considering different transportation companies which aim to collaborate on the transportation of goods and gain benefits regarding costs and emissions. The companies act as the DMs, and the problem and knowledge about the situation are used in solving the problem. The knowledge about which kind of solution is desired, called here as predefined desirability of solutions, can be used to modify the aggregation of multiple preferences to the collective preference. The authors present different methods using different predefined desirability of solutions.

Papers relying on the SDM to form the collective preference are discussed next. The SDM can use different decision making and decision analysis tools to help make an informed judgment, but eventually, this approach relies on the SDM's expertise. A stakeholder analysis supports the SDM in [17]. The SDM determines the importance of the individual DMs through an influence and interest ranking method analysis. The rankings of the importance of the DMs are converted to weights. A collective weight vector is formed from the DMs' preferences and the degrees of importance of the DMs.

Direct approaches

Utilizing a direct approach in the first stage of a two-stage algorithm in [11] where the DMs provide their preferences for the objectives using weights. The method includes an SDM. The weighted sum method using the weight vectors generates a set of alternative solutions. At second stage of the algorithm the DMs evaluate the generated alternative solutions independently without negotiations with other DMs and score each alternative solution. The SDM sets the degrees of importance of individual DMs with weighting coefficients.

Next, we discuss a priori EMOAs with a direct approach. Direct preference incorporation utilizes multiple preferences without aggregation; hence, multiple solutions will be found even in scalarization-based methods. Generally, a priori methods incorporating a direct approach in the reviewed literature try to find solutions that differ a lot from each other or on the opposite, solutions that are similar. The differing solutions aim to provide more options for the DMs, while similar solutions try to make the selecting final solution easier.

The following two papers consider a resource allocation problem, where non-finite resources are allocated between all the DMs participating in the process. The goal is to allow the DMs to enter their own preferences to integrate all of the DMs' preferences into the search for a solution to achieve the best objective values for each DM. The reference points the DMs provide are used directly to determine the level of satisfaction of a given solution in [72]. The evolutionary method evolves potential solutions by alternating between the evolution of the solutions and the evolution of the preferences. As a result, a set of solutions is found. Following this approach in [73], the reference points given by the DMs are used as weights for the objectives. In this way, the weighted sum based scalarization is used as a fitness function in the evolutionary method. Multiple solutions are found as a result.

Another a priori evolutionary approach to the incorporation of multiple preferences in a direct manner is presented in [47]. The algorithm uses weight vectors given by the DMs to find multiple relevant regions of interest to multiple DMs simultaneously and multiple non-dominated solutions in those regions of interest. The crowding distance operator in NSGA-II is replaced with a preference measure. The preference measure prefers the solutions closer to the Pareto optimal solutions corresponding to the weight vectors in the objective space [47]. To maintain diversity, the solutions close to this preferred solution are penalized in the preference measure.

Direct approaches with a consensus measure

As defined earlier, a consensus solution refers to a solution that all or most of the DMs can accept. The following method utilizes a consensus measure based on the distances between the DMs' preferences. Authors in [64] present a rank based consensus approach to find a set of consensus solutions utilizing R-NSGA-II, which is a variation of NSGA-II that utilizes a reference point as the preference information. In the method, the crowding distance operator of R-NSGA-II is modified by changing how the ranking is calculated and how the solutions in the same rank are preferred. The method minimizes normalized distances between the reference points of DMs and the non-dominated solutions. The method is complemented by an additional ranking scheme, which removes the effects from the differences in the magnitudes of the objectives.

In the following, we introduce distance based consensus measures. These approaches consider fuzzy preferences and the robustness of preferences and solutions and focus on fuzzy GDM-MOPs. The methods focus on calculating the desirability of solutions with specific measures. The aim is to find a set of consensus solutions incorporating the DMs' preferences and taking the robustness of solutions into account.

In [89], the original MOP is reformulated to a biobjective optimization problem minimizing a consensus and a preference robustness measure. The consensus measure is based on the distances between the DMs' fuzzy reference points and the distance of the solution to a collective preference. The collective preference is the mean of the reference points. The robustness measure is defined to evaluate the effect of small perturbations in the objective space of the MOP. The DMs' fuzzy reference points are given to modified NSGA-II to find a set of solutions, which are measured in the new consensus robustness space. A set of consensus solutions is reached by solving this biobjective problem using NSGA-II with a modified ranking procedure to incorporate the DMs' preferences and the degree of importance of the DMs.

Inspired by the work done in [89] and also using fuzzy reference points, the authors in [58] redefine consensus and introduce a new measure called robust consensus combining the redefined consensus measure and the preference robustness measure. In [57] the authors introduce a total of twelve robust consensus measures. The different robust consensus measures are formed with different aggregation operators. As mentioned the DMs' preferences are given as either fuzzy reference points or as triangular or Gaussian membership functions. In the case of fuzzy reference points, the preferences are transformed into triangular or Gaussian membership functions. In both of the methods discussed, the considered MOP is reformulated by including the robust consensus measure as an additional objective. A set of robust consensus solutions is found by solving the reformulated problem using EMOAs in [57, 58]. The degree of importance of the DMs can be specified in the methods by using weights.

4.1.3 Selecting the final solution

This subsection discusses the approaches taken to select the final solution in a priori GDM-MOO methods. As mentioned, the goal of solving a GDM-MOP is to find the final solution to be implemented. A straightforward approach to find the final solution would be to form a collective preference and use any preference based MOO method, that provides a single Pareto optimal solution and set that as the final solution. However, selecting the final solution is not popular in the reviewed literature as Table 3 suggests. Instead, the focus is on finding collective solutions. This can be achieved by using the collective preference in any MOO method that provides multiple solutions, or the preferences of the DMs are used directly to provide a set of collective solutions. Selecting the final solution from the collective solutions ought to be easier. The final solution selection requires utilizing some group decision method, and these group decision processes are rarely demonstrated in the reviewed literature.

Next, the methods that provide the means to select the final solution are discussed. These methods utilize a collective preference in different scalarization-based methods. The weighted sum method is used in [11, 32, 50] to find the final solution using the collective weight vector. Applying another scalarization-based method, the WIN method presented in [17] chooses the final solution that minimizes the scalarized WIN value to a normalized ideal vector.

In [28], the fuzzy outranking-based method returns only the non-dominated solutions that satisfy an indicator function considering the number of satisfied DMs. In the cases, where there are more than one solution, the group (or the SDM) apply any group decision method that has been agreed by the group.

Most a priori methods in the reviewed literature do not select the final solution, see [9, 10, 31, 47, 48, 57, 58, 64, 72, 73, 80, 89]. Instead, they focus on finding a set of collective solutions from where the DMs are expected to select the final solution either by negotiations [31] or by using some new information e.g., DMs' preferences of consensus and robustness measures like suggested in [89].

4.2 A posteriori methods

In a posteriori methods, an (approximation of the) set of Pareto optimal solutions is generated before DMs are included in the solution process. The DMs need to evaluate (a subset of) the solutions and select the final one. Hence this section does not follow a similar structure as the others. First, in Subsection 4.2.1, we consider the MOP formulation stage and how the

Pareto set is found. Then, in Subsection 4.2.2, we focus on the group decision making phase of selecting the final solution.

Papers where a posteriori methods have been proposed are summarized in Table 4. In Table 4, the reference column indicates the paper under consideration on that row. The preference information types column mentions the used preference information types separated by a comma. If the DMs had options the word *or* is separating the options. The group decision method(s) used indicates the types of the group decision methods utilized to determine collective solutions. The final solution column indicates whether the method is able to identify the final solution for the GDM-MOP.

Reference	**Preference information types**	**Group decision method(s) used**	**Final solution**
[30]	ranking solutions	SC, FB	Yes
[60]	ranking solutions	SC, FB	Yes
[1]	ranking solutions	SC, FB	Yes
[2]	ranking solutions	SC, FB	Yes
[66]	ranking solutions with impasse	FB	Yes
[65]	ranking solutions	SC	Yes
[56]	ranking solutions, least preferred solution	SC, FB	No
[78]	most preferred solution	score voting	Yes
[5]	pairwise comparison of objectives	group discussion	Yes
[33]	weights, pairwise comparisons of objectives, range of feasible solutions	SC	Yes

Table 4. The reviewed a posteriori GDM-MOO methods in a table format.

4.2.1 Finding solutions

In the papers where a posteriori methods were proposed many authors considered real-world problems. Hence, the developed methods are typically tailored to these specific problems. These methods include a lot of problem-specific information and assumptions that are not relevant to the focus of this review and therefore are not discussed in much detail. Instead, we focus on how the MOP was solved.

In the following, we discuss the problem formulation stage of a posteriori GDM-MOO methods. In [1, 2, 30, 56, 60, 65, 66], the authors consider real-world problems that utilize different simulators and MOO methods aiming to find good solutions for policymakers. The GDM-MOP considered in [30] involves water allocation in river systems, irrigation networks in [56], water distribution system of a city in [60], infrastructure planning regarding urban storm water management in [65] and simulation models regarding groundwater contamination in [1, 2, 66].

The DMs considered in these problems are stakeholders involved in some way, e.g. farmers of an area considered. The primary purpose of these studies is to build a simulation-optimization model considering the DM's preferences and the uncertainties related to the problem and to find collective solutions. From the collective solutions, a final solution can be selected. Generally, each DM only focuses on one objective function, and the DMs' preferences of the objective functions to optimize are either already given or for example, as in [1], the DMs' objective functions and preferences are identified through data analyses and interviews.

The objective function that is important to an individual DM is modeled as that DM's utility function in [1, 2, 30, 65, 66]. The optimization stage starts after the problem has been formed. The authors use NSGA-II to generate a set of non-dominated solutions in [1, 2, 30, 56, 60, 65, 66]. In some of the papers, the generated set of solutions is related to a specific (uncertainty) scenario, leading to a different set of solutions for each of the scenarios considered. Since each of the scenarios require a single final solution, the different scenarios are not considered further in this review. Instead, we focus on the set of solutions found and what GDM methods are used to select the final solution.

Next, we discuss other papers utilizing a posteriori methods. In [78], the authors design a collaborative group decision making shipping marketplace without a moderator using Blockchain. The three objectives functions are indicators relating to maximizing shipping numbers and values and minimizing empty trucks. The DMs generate Pareto optimal solutions by solving the MOP with different scalarization methods.

Considering the trade-offs between logistic costs and response times, the location-allocation problem of placing temporary relief distribution centers is presented in [5]. The objective functions are to minimize response time, logistic costs and unsatisfied demands. The humanitarian response experts took part via an online questionnaire to elicit preferences regarding the weights of the objectives and to provide pairwise comparisons of the objectives. The preferences of the experts and a Monte Carlo simulation are used to generate different weight vectors. A problem-specific weighted sum approach using the generated weight vectors is used to generate a set of solutions.

In [33], the authors consider a real-world problem optimizing public transportation networks. The objective functions are to maximize total travel time savings, maintain balanced origin and destination terminals and to minimize the construction budget. The MOP is solved with an EMOA, and a set of non-dominated solutions is found. In the next subsection, we consider how the DMs provide their preferences for the generated solutions.

4.2.2 Selecting the final solution

As mentioned, in a posteriori methods, a set of Pareto optimal solutions is found, and the DMs need to evaluate (a subset) of them. Hence, this subsection discusses what type of preference information the DMs give about the solutions shown to them and how the different preferences of solutions are aggregated for final solution selection. Various types of preferences can be given. In the reviewed literature regarding a posteriori GDM-MOO methods, most of the methods rely on voting-based or negotiation-based social choice methods or methods from the MCDA literature. Table 4 summarizes the preference information types given to the methods and the methods used in the final solution selection.

Social choice methods

In the following discussion, we consider the papers that use social choice methods to select the final solution. In [1, 2, 30, 56, 60, 65, 66], the DMs' preferences are modeled with utility functions and then different social choice methods are used to rank the solutions. In these papers, as mentioned, the authors often present several final solutions concerning alternative solutions regarding different scenarios or uncertainties in the problem and by using multiple different social choice methods. We consider how the final solution is selected from the best solutions (according to the social choice methods). The social choice methods discussed in this subsection can be split into voting-based social choice methods like Borda count and negotiation-based methods like unanimity fallback bargaining as introduced in Section 3.3.

The most popular social choice methods in the reviewed literature are Borda count [1, 2, 30, 56, 60, 65] and median voting rule [1, 30, 56, 60]. The papers utilize at least four different social choice methods. In [56], the DMs rank the solutions and their least preferred solution and they are used as an input for social choice methods and for a game-theory based bargaining method, respectively.

Next, we discuss how the final solution is selected. The steps involving the DMs and how the final solution is selected were not always clear in the papers. Hence, the following is our interpretation of the approaches to select the final solution.

First, the DMs rank the solutions using the social choice methods in [30, 60]. Then, instead of using a fallback bargaining method to determine the final solution directly, the DMs bargain over the social choice methods. The best social choice method according to the fallback bargaining, determines the final solution. In [60], the best social choice methods, according to the scenario considered, are Borda count, Condorcet's practical method and plurality rule. The corresponding solutions are selected as the final solutions per the scenario considered. In [30] the unanimity fallback bargaining method results in the Borda count and Condorcet choice as the best social choice method in two different scenarios.

Another approach in the reviewed literature is to use fallback bargaining side by side with the social choice methods to rank the solutions. In [1], the final solution is the one that both of the fallback bargaining methods suggested, while in [56] the authors use several social choice methods, and fallback bargaining methods to find a set of solutions but do not indicate which is selected as the final solution.

The fallback bargaining method can be also the only social choice method to select the final solution. In [66], the fallback bargaining method is implemented with an impasse. The method is able to find the final solution in each scenario considered. In [65], social choice methods are used without using any fallback bargaining methods. The Borda count method is determined to be the best social choice method and decides the final solution.

Other methods

An online questionnaire is used to get preference information from DMs in [33]. The DMs provide weights for objective functions and pairwise comparisons of the objectives for three different MCDA methods including AHP. The DMs indicate how many solutions they are willing to rank, and then rank them using the MCDA methods. In this way, each DM has a ranking of the solutions for each of the MCDA ranking methods used. Then, the rankings are given to the Borda count method to obtain the final combined ranking. The highest-ranked solution is selected as the final solution.

In [5], the authors test the method with real DMs who are experts in the humanitarian response. The eight DMs are split into two test groups with four DMs, each group solving the problem independently. A moderator supports the DMs and ensures that everyone clearly understands the decision situation. The DMs are provided with generated solutions and some visualizations of the solutions. The DMs had one hour to discuss and converge on the final solution. Both of the subgroups did agree on the final solution.

As mentioned, in [78], different scalarization-based methods are used to generate a set of Pareto optimal solutions representing collaborative shipping contracts. After the solutions have been found, the DMs perform a score voting on the solutions. The solution with the highest score is selected as the final solution.

4.3 Interactive methods

In this section, we discuss interactively given preference information and how it is incorporated into GDM-MOO methods. In Subsection 4.3.1, we consider what kind of preference information is asked from the DMs. In Subsection 4.3.2, we discuss how the multiple preferences are incorporated in to the solution process in an interactive manner. Lastly, in Subsection 4.3.3 we consider how the final solution is selected.

In Table 5, we summarize the papers proposing interactive methods for GDM. The reference column indicates the paper referred to as on that row and the preference information types column mentions the used preference information types separated by a comma. If the DMs had options, the word *or* is separating the options. The second column indicates whether the methods used indirect or direct approaches. The third, final solution, column indicates whether the method can identify the final solution for the GDM-MOP.

Reference	**Preference information types**	**Incorporating multiple preferences**	**Final solution**
[85]	binary satisfaction to solution	direct	Yes
[29]	interval weights, veto, majority, credibility thresholds	indirect	Yes
Mhab-EC [76]	weights for objectives	direct	No
[46]	preference ranking	indirect	Yes
CIMO [7]	classification of objectives	direct	Yes
[62]	-	direct	Yes
Mhab-EC [77]	weights for objectives	direct	No
CI-NSGA-II, CI-SMS-EMOA, CI-SPEA2 [18]	pairwise comparison or free design in game	indirect	No
NEMO-GROUP [42]	pairwise comparison of solutions	indirect	No
[87]	-	direct	No
[6]	interval weights, veto, majority, credibility thresholds	indirect	Yes
[27]	interval weights, veto, conservatism, credibility and majority thresholds	indirect	Yes
CI-NSGA-II, CI-SMS-EMOA [19]	pairwise comparison of solutions	indirect	Yes

Table 5. The reviewed interactive GDM-MOO methods in a table format.

4.3.1 Preferences in interactive methods

This subsection focuses on the types of preference information the DMs provide in the different interactive GDM-MOO methods. The reviewed literature contained various preference articulation styles, collected from the DMs in different stages of the decision making process. As mentioned, Table 5 summarizes the different preference information types used in interactive methods.

To start with, we discuss interactive methods utilizing evolutionary algorithms with different preference information types. The preferences include weights for objective functions and pairwise comparisons of solutions. After this, we present different scalarization-based algorithms with varying preference information types.

Weights for objective functions are commonly used in interactive GDM-MOO methods. Considering precise preference information, the DMs provide weights for objectives in [76, 77]. The weight vector is given to an agent representing the DMs' interests in an agent-based framework, called multi-human-agent-based evolutionary computation (Mhab-EC). The Mhab-EC uses evolutionary methods to generate non-dominated solutions that are shown to the DMs. After seeing the generated non-dominated solutions, the DMs can provide new weights for restarting the method and reach different solutions. Mhab-EC is meant to be run several times with DMs adjusting their preferences while gaining new information from the solutions found. Hence the authors call their method interactive.

The following papers focus on situations where the DMs' preference information can be imprecise or imperfect and the objective values may be uncertain. The DMs can adjust their preferences during the consensus reaching processes led by the SDM in these methods.

The DMs expressed their preferences as interval weights to model the imprecise information in the DMs' preferences in [6, 27, 29]. Compensatory preference models mean that poor values in some objectives can be compensated by good values in others, while non-compensatory preference models do not allow this. Depending on whether a DM wants to express compensatory or non-compensatory preferences, the DM chooses which preference model to follow in [6, 29], either a weighted sum model or an interval outranking model, respectively. In the weighted sum model, the DM needs only to provide the interval weights for objectives. Next, we discuss the case when the DM wishes to express non-compensatory preferences.

Influenced by multi-criteria ordinal classification methods [6, 27, 29], in the interval outranking model, interval weights are supplemented with additional parameters from the DMs. These parameters included veto, majority and credibility thresholds in [6, 27, 29]. Veto thresholds allow considering veto situations in the outranking model. Majority and credibility thresholds are used in the outranking model to establish precise preferences from imprecise preferences. Besides these parameters, in [27], the authors include conservatism thresholds to model the risk-taking attitudes of different DMs regarding the objective values. The conservatism threshold is utilized in the outranking model in handling imprecise preferences.

Pairwise comparisons of solutions are another common type of preference articulation in interactive methods, used in [18, 19, 42]. The interactive methods used evolutionary algorithms utilizing pairwise comparisons of solutions and aggregating the preference information to a collective preference to guide the search process. In [18, 19], the DMs perform pairwise comparisons of randomly selected non-dominated solutions. The DMs select their best solution from the shown solutions. Additionally, in [18], the DMs can express their preferences by a free design mode in a gamified graphical user interface dedicated to the facility location problem in a video game. This is discussed in more detail in the following subsection.

In [42], the authors propose a set of evolutionary algorithms called NEMO-GROUP, where pairwise comparisons of solutions of several DMs are incorporated into the evolutionary search. At regular intervals, each DM compares a pair of randomly selected non-dominated solutions. The best solutions selected are used to form utility functions representing the DM's and the group's preferences.

Next we discuss miscellaneous types of preference information in interactive methods. The DMs can indicate their preferences at each iteration of the algorithm by solving a subproblem from which the values of a utility function can be deduced in [62]. The paper did not explain how an individual DM can solve this problem. The DMs select the most preferred solutions of the shown solutions in [87]. The median of the preferences is computed, and the solutions adjacent to this median determine the direction the algorithm moves in the objective space. In [7], the DMs classify the objective functions using natural language into five different classes. The authors in [85] consider a real-world problem optimizing a wine-harvest schedule. There are two DMs and two objective functions. The DMs do not give any preferences before the optimization, only after a solution is generated and shown to them. The DMs indicate their agreement or disagreement of the suggested solution.

The interactive method presented in [46] generates a set of Pareto optimal solutions using scalarization-based MOO methods. The DMs must decide how many solutions they want to see after each iteration. The DMs discuss in a group about the solutions and form tentative preference rankings. After the group discussion, the DMs provide ordinal rankings of the solutions independently without group interaction. The ordinal rankings are aggregated to a collective preference ranking used in the optimization.

4.3.2 Using multiple preferences in the solution process

In this subsection, we consider how the multiple preferences (from several DMs) are incorporated in the interactive GDM-MOO methods. The preference information is expressed progressively during the solution process. The preferences can be incorporated in either an indirect or a direct manner. If group interaction is possible, the DMs can take part in a consensus reaching process to bring their preferences closer to each other. The consensus reaching process can also be used after finding solutions to help with final solution selection.

Indirect approaches with evolutionary algorithms

Next, we discuss interactive GDM-MOO methods that involve an indirect approach and study how multiple preferences are combined to collective preference(s) and utilized to guide the solution process. In [18, 19], in each iteration the DMs perform pairwise comparisons of two randomly selected solutions. The DMs select the best solution and then all of the pairwise comparisons are used to form a collective reference point. The selection operators of the developed evolutionary algorithms are replaced by a selection operator based on the distance to the collective reference point. The variation operator is replaced in [18] by implementing a free design mode, where the DMs can update and redesign the game objects, like trucks and warehouses, in a dynamic manner. The redesigned solutions are re-injected into the population.

The DMs make pairwise comparisons of solutions in the NEMO-GROUP methods in [42]. The collective preference model combines individual pairwise comparisons into a comprehensive value representing the whole group of DMs. The authors form two types of group utility functions, which evaluate the solutions. The NEMO-GROUP methods also incorporate weights for DMs' degrees of importance. The different utility functions are used in the NEMO-GROUP methods to find a set of consensus solutions.

Indirect approaches with a consensus measure

Methods to be discussed in the following build a collective preference model to measure the consensus of the DMs about solutions. The methods consider imprecise preference information and groups, where interaction and communication is possible among the DMs. In addition, these methods assume that there is a moderator (or an SDM acting as a moderator) available to guide the consensus reaching process. The consensus measure counts the number of DMs who are satisfied and respectively dissatisfied with their current objective values in [27, 29]. In [6], the consensus measure considers the number of highly satisfied, satisfied, dissatisfied and highly dissatisfied DMs. In the methods in [6, 27, 29], it is assumed that with the guidance of the moderator, the DMs are able to set up their individual MOPs. The methods start with a consensus reaching process guided by the moderator. The consensus reaching process aims to make the DMs' preferences more similar. Then, the DMs solve the MOP individually by using any MOO method. The DMs distinguish their best solution and those solutions are considered as reference points for the DMs in the following GDM-MOP.

All of the methods in [6, 27, 29], use EMOAs incorporating the collective model to find non-dominated solutions which are then evaluated by the consensus measure as discussed. The solutions found are consensus solutions. If no solutions are found, another consensus reaching process must be performed, and after the DMs' preferences have been adjusted, the optimization is performed again.

Here, a preference model refers to either an interval outranking model or an weighted sum model with interval numbers. The DMs set up the parameters for the preference model they prefer to use. The parameters for the interval outranking model included veto, majority and credibility thresholds and both preference models required DMs to specify interval weights for objectives. In addition, in [27] the DMs need to specify a conservatism threshold. In [6, 29] the DMs are split into subgroups corresponding to the preference model for the subsequent consensus reaching processes.

Next, we discuss how the interval outranking-based methods proposed in [6, 27, 29] handle measuring the consensus degree of solutions. Some additional information is needed. In [29], each DM sets a boundary for each objective function to classify any solution into a set of satisfying or dissatisfying solutions. This boundary should not be shared with other DMs. In [6], the DMs set their representative solutions by classifying some solutions to either satisfactory or unsatisfactory ones. Representative solutions mean examples of satisfying and unsatisfying solutions. It is assumed that the DM has at least a vague idea of what makes a solution satisfying and what makes a solution unsatisfying. Both the boundary, and the representative solutions, are used in the preference model to classify solutions to in satisfactory or dissatisfactory classes. In this way, the consensus degree of given solution can be computed.

The SDM's role is important in the two methods presented in [27]. In the first method, the SDM creates a collective model aggregating all the DMs' preferences. As mentioned, each DM expressed interval weights for objectives, reference point and veto, majority, credibility and conservatism thresholds. The collective model is updated by the SDM after each consensus reaching process and can be used to find the final solution. In the second method, the SDM focuses on the consensus measure maximizing group satisfaction and minimizing dissatisfaction. The solution close to the individual DM's preferences is considered to satisfy that DM. The concept of closeness is defined in the method. However, the SDM reserves the final judgment to define the concept of closeness and other parameters of the model, including the degrees of importance of DMs. After building the collective model, the SDM solves the problem by looking for consensus solutions that maximize group satisfaction and minimize group dissatisfaction. If group interaction is possible, a consensus reaching process under the moderation of the SDM can be performed to make the preferences of the DMs more alike.

Indirect approach with a scalarization-based method

Next, we discuss a scalarization-based method that uses an indirect approach, including group discussions, to help achieve a consensus ranking of the DMs' rankings of solutions. The method in [46] used two multiobjective linear optimization methods to generate a set of Pareto optimal solutions. To use these methods with multiple DMs, the DMs' preferences are aggregated into a collective preference in the following manner. The DMs discuss the generated Pareto optimal solutions and each provides a tentative ordinal ranking of the solutions. Then the DMs can adjust their ordinal rankings of the solutions without group interaction. These will be converted to equivalent cardinal rankings, and then a collective ranking is formed. Based on the collective ranking, the method generates four new solutions, which the DMs again discuss. The DMs can decide to restart the process or decide on the final solution.

Direct approaches

In the following, we discuss methods utilizing a direct preference information approach. Mhab-EC aims to find consensus solution(s) without needing face-to-face meetings of DMs. As mentioned, in Mhab-EC [76, 77], the DMs give weights for objectives to be used in agents as the agent's preferences. The agents work in a virtual agent space, move randomly and share their preferences with other agents. The sharing mechanism makes the individual agents' preferences more similar to the average of the nearby agents. The agents generate new preferences at each generation. In [77], the authors extend Mhab-EC to let the DMs adjust their preferences between iterations of the method. The solutions found after a maximum number of iterations is reached are defined as the consensus solutions.

The DMs provide their preferences by classifying the objective functions in [7]. The method is set up to divide a MOP into subproblems that the different DMs solve in parallel. The SDM is supervising the DMs and after each DM has solved their subproblems, the SDM solves their MOP considering the (not necessarily Pareto optimal) solutions found by the individual DMs. This is repeated until a stopping criterion is reached. The authors in [62, 87] assume that the preference structure of each DM can be represented by a utility function. In the methods, the DMs' utility functions are used to find collective solutions.

4.3.3 Selecting the final solution

As mentioned, the goal in solving a GDM-MOP in practice is to find the final solution to implement. In the interactive GDM-MOO methods, various ways are utilized to select the final solution. Some methods include a consensus reaching process with a moderator, some methods do not involve a moderator in this, some methods utilize different voting approaches, and some rely on the SDM's expertise to select the final solution. In the following, we discuss how the final solution is selected.

As mentioned, in [6, 27, 29], a set of consensus solutions is found and the authors propose separate ways of then selecting the final solution, depending on the number of solutions in the set of consensus solutions. If there is a single consensus solution, it is selected as the final solution. Otherwise, in [27, 29] the SDM selects a group decision method to select the final solution. In [29], the group decision method is based on the outranking credibility index of a given solution. The solution with the best value according to this indicator is selected as the final solution. In [27], either the SDM selects the final solution or the group decision method is a consensus measure to calculate the satisfaction of the DMs with the consensus solutions. The solution with the best consensus degree is selected as the final solution. The authors in [6] consider two other approaches to selecting the final solution from multiple consensus solutions. In case of two consensus solutions, the group selects the final solution by voting. If there are three or more consensus solutions, the final solution is chosen by a Borda count (the solution with the highest score).

Next, we discuss methods where a consensus reaching process is used to determine the final solution. In [46], the group discusses the four consensus solutions (obtained so far) according to the collective ranking of the DMs. The DMs have to agree on the final solution selected from the consensus solutions. Otherwise, the group decides whether to continue the search or end the process by selecting the final solution with a majority vote. The negotiation protocol considering two DMs in a wine-harvesting schedule problem introduced in [85], proceeds as follows until an agreement is found. The authors utilized an approach to find the starting Pareto optimal solution for the negotiation by minimizing the augmented weighted Chebyshev distance to the ideal vector. The starting Pareto optimal solution is shown to the two DMs, and if both DMs agree that this solution can be selected as the final solution, the negotiations end. Otherwise, a new solution is generated by exploring the neighborhood around the solution last shown to the DMs. The new solutions shown are solutions that increase one objective in a similar percentage to the decrease of the second objective. When both DMs give their agreement on the solution generated, the process ends by selecting that solution as the final one.

Next, we consider methods using voting-based or MCDA-based approaches to select the final solution. In [62, 87], majority voting is used to select the final solution and in [7], the AHP method is used to select the final solution among the different generated solutions. In the gamified decision making process in [19], at the end of the game, the method has created only one solution, and that is presented as the final solution. Since, the DMs' preferences are taken equally into account, that solution is best according the majority of the group.

In the following methods, the focus is not on final solution selection. Generally, the found solutions are called collective ones, since they are found using multiple preferences (from the group of DMs) or they are called consensus solutions, since they are evaluated using a consensus measure. In [42, 76, 77] the goal of the solution process is to find a set of consensus solutions and in [18], to find a set of collective solutions. As mentioned, in this review, we refer to collective solutions, when solutions are found (in any way) using the multiple preferences of the DMs. Also, we refer to consensus solutions as solutions that satisfy a consensus degree. However, the authors in [76, 77] define the converged solutions as the consensus solutions. In the case of multiple consensus solutions, it is suggested that a group decision method is needed to select the final solution. In the end, the final solution is not selected.

5 Discussion

In this chapter, we discuss several important aspects of the reviewed literature regarding GDM-MOPs and GDM-MOO methods. The focus of the literature review was to examine the state of the art of the combination of GDM and MOO methods. The reviewed literature consists of 40 papers and in the following, we summarize the findings regarding the research questions, which are discussed in Chapter 4. The first research question asks what kinds of preference information types the DMs were expected to express. The second research question asks how the multiple preferences from several DMs were incorporated into the solution process. The third research question asks how the final solution was selected in GDM-MOPs.

To answer the research questions, we evaluated different preference information types given by multiple DMs and how the preferences were incorporated into the solution process. We considered how the methods supported the DMs in the final solution selection. We adopted the taxonomy from the MOO field to classify GDM-MOO methods based on the timing of incorporating the DMs' preference information in the solution process. In addition, we introduced indirect and direct approaches to including the multiple preferences of several DMs in the solution process.

First, in Section 5.1, we discuss the distribution of the papers according to the performed classification and suggest desirable properties of GDM-MOO methods. The desirable properties are formed by combining the relevant findings from the reviewed literature with the general issues to consider when solving GDM problems. In Section 5.2, we discuss these desirable properties in light of the results of the literature review. Lastly, in Section 5.3, we consider the limitations of the study and discuss future research topics.

5.1 Desirable properties of GDM-MOO methods

To start with, let us consider the distribution of the papers according to the performed classification. In Figure 4, most of the methods proposed are a priori methods followed by interactive methods and then a posteriori methods. However, we do need to remind, that one of the interactive methods (utilized in two papers) could have been classified as an a priori method. In this thesis, the authors' own classification of their method belonging to interactive methods was respected.

Considering the papers using a posteriori methods, it is surprising that it has the smallest amount of papers, since GDM in a posteriori MOO methods is the most similar to the extensive GDM research in MCDA problem settings. The reason why the case is this comes from the way this literature review was conducted. As mentioned in Chapter 4, if the paper considers an MCDA problem setting, with an explicitly given set of alternative solutions, the paper was excluded since it did not have an optimization perspective where solutions are generated from the feasible set.

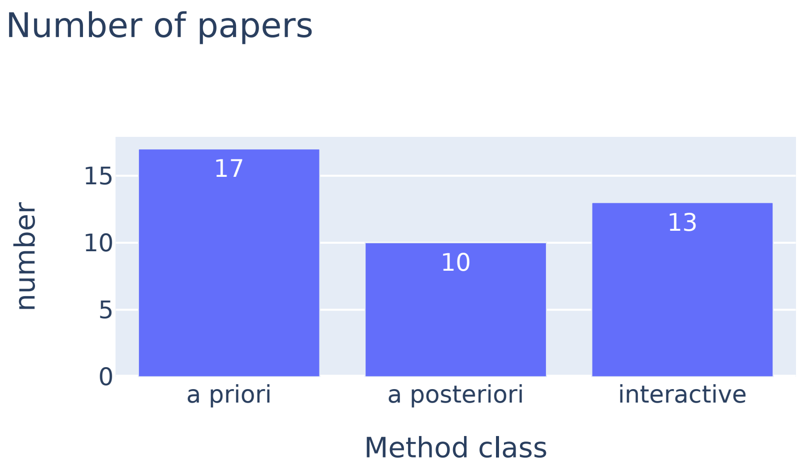

Figure 4. Number of papers reviewed in the method classes

Desirable properties of GDM-MOO methods

The way of developing GDM-MOO methods in the reviewed literature and how they are discussed raises questions and concerns. Much of the information and descriptions of the methods are not compatible. For example, how can a method be said to solve a GDM-MOP if the final solution is not selected? Furthermore, many GDM-MOO methods lack testing with real DMs in real-world MOPs.

The following properties are derived from these questions and the results achieved by performing this literature review. The properties are purposed to indicate important aspects of solving a GDM-MOP, assumptions or elements of the DMs and their preference information that should be considered in solving a GDM-MOP or in developing a GDM-MOO method. The properties aim to ensure that the important aspects of solving GDM-MOPs are addressed and the methods will have validity in real-world problems. Therefore, we list desirable properties to aim to meet so that GDM-MOO methods would be more applicable to real-world problems and easier to understand and apply in practice.

- Desirable properties (DP)

  DP1 - The method is tested with real DMs.

  DP2 - The method is tested with real-world MOPs.

  DP3 - The authors report for which kind of groups the method is developed for.

  DP4 - The preference information the DMs are required to provide is understandable for them.

  DP5 - The method incorporates the preferences of each of the DMs.

  DP6 - The method provides support to the final solution selection.

  DP7 - Most (if not all) of the DMs can accept the final solution.

Next, we briefly discuss the desirable properties. DP1 and DP2 are often entangled together. DP1 requires that the DMs are domain experts in the problem and not e.g. the authors acting as DMs. DP2 requires that the GDM-MOO method is tested with real-world MOPs instead of benchmark problems, where the objective functions do not mean anything to the DMs. The purpose of DP1 and DP2 is to ensure that the GDM-MOO method will work in real-world problems with real human DMs. Fulfilling DP1 and DP2 brings validity to the usability of the developed method in solving real-world MOPs.

The purpose of DP3 is to point out that different types of groups will benefit from different types of GDM-MOO methods. Authors should communicate to which types of groups the method is developed for, so it can be easily applied by others in solving suitable GDM-MOPs with similar types of groups. Defining the group structure, e.g. in a manner discussed in Section 3.1, will make sure the relevant aspects are communicated (and therefore considered in the method development).

DP4 requires that the preference information type is such that each DM can understand it. At the same time, the DMs should not have to provide too many different types of preferences, since that also affects the understandability and may increase cognitive load.

DP5 indicates that the preferences of each DM are incorporated in the GDM-MOO method. The preferences of the DMs should guide the search and it would be preferable that all the DMs' preferences would be considered. However, DP5 does not require that each of the individual DM's preference will be considered in a similar manner or with a similar effect in the solution process.

DP6 ensures that the GDM-MOO method supports the DMs in selecting the final solution. If the final solution is not selected, there is no solution to implement in practice and the GDM-MOP has not been solved. The GDM-MOO method should be able to help the DMs to find collective solution(s) from which the DMs are able to determine the final solution.

The purpose of DP7 is to bring some validity to the acceptance of the final solution. As discussed, by using e.g. aggregation-based methods, the solution(s) found may be too different from the solution(s) an individual DM would have preferred. This may lead to a situation where the final solution is not implemented in practice. The GDM-MOO method should contain some ways to guarantee the acceptance of the final solution or the authors should e.g. directly ask from the DMs about the acceptance of the final solution.

How are (or the lack of) the desirable properties manifested in the reviewed literature? Are there some findings that should be considered in future research? In the next section, we discuss these desirable properties and the lack of fulfilling them in the reviewed literature.

5.2 Evaluating the reviewed literature

Here, we discuss the desirable properties in the context of the reviewed GDM-MOO literature. We highlight papers, where the desirable properties are fulfilled and discuss papers where they are ignored. Table 6 gives an overview of how the desirable properties DP1, DP2, DP3 and DP7 are met in the reviewed papers. We explain the entries and their meaning in the table, with respect to the desirable property under consideration. We discussed DP4, DP5 and DP6 in Chapter 4, hence they are not repeated in the table.

Desirable properties	**DP1**	**DP2**	**DP3**	**DP7**
[72, 73]	✗	✗	C_d, I_n, DI_e	✗
[50]	✓	✓	$-,-,DI\_n$	✗
OptMPNDS, OptMPNDS2 [48, 80]	✗	✗	$-,I\_y,DI\_e$	✗
[11]	✓	✓	C_d, I_n, DI_n	✓
MWS-NSGA [47]	✗	✗	$-,-,DI\_e$	✗
[31]	✗	✓	C_d, I_n, DI_n	✗
[64]	✗	✓	$-,-,DI\_e$	✗
[57, 58, 89]	✗	✗	$-,-,DI\_n$	✗
WIN [17]	✓	✓	$-,-,DI\_n$	✓
NSS-GPA, W-NSS-GPA [9, 10]	✗	✗	C_a, I_y, DI_n	✗
[28, 29]	✗	✓	C_a, I_n, DI_n	✓
[1, 2, 30, 32, 33, 60, 65, 66, 85]	✓	✓	$-,-,DI\_e$	✓
[56]	✓	✓	$-,-,DI\_e$	✗
[78, 87]	✗	✗	$-,-,DI\_e$	✗
[5]	✓	✓	$C\_a,-,DI\_e$	✓
Mhab-EC [76, 77]	✗	✗	$C\_d,-,DI\_e$	✗
NEMO-GROUP [42]	✗	✗	$-,-,DI\_n$	✗
[46]	✓	✓	C_a, I_y, DI_e	✓
CIMO [7]	✗	✓	$-,I\_n,DI\_n$	✗
[62]	✗	✗	$-,-,DI\_e$	✓
CI-NSGA-II, CI-SMS-EMOA, CI-SPEA2 [18]	✓	✓	$-,-,DI\_e$	✗
CI-NSGA-II, CI-SMS-EMOA [19]	✓	✓	$-,-,DI\_e$	✓
[6]	✗	✓	C_a, I_y, DI_e	✓
[27]	✗	✓	$C\_a,-,DI\_n$	✓

Table 6: The desirable properties DP1, DP2, DP3 and DP7 in the reviewed literature.

Discussion of DP1 and DP2

In the following, we discuss DP1 and DP2 in light of the reviewed literature. Let us get started with how to decipher Table 6 regarding DP1 and DP2. The symbol ✓ indicates that DP1 has been fulfilled in the corresponding paper and the symbol ✗ indicates the opposite. In this review, we interpret that DMs are real-world DMs, if the authors clearly indicate this, e.g. by reporting the roles of the DMs. For example in [85], the authors reported that one of the DMs is an oenologist (wine-harvesting expert) and the other is a field manager (economical expert). We regard also students who are testing GDM-MOO methods as real-world DMs. The cases, where the authors are acting as DMs have been marked with the ✗. In some of the papers, it was very hard to deduce whether real DMs were involved or the authors acted as DMs. In these cases, it is assumed that there were no real DMs involved.

Regarding DP2, the symbol ✓ indicates that DP2 has been fulfilled in the paper and the symbol ✗ indicates the opposite. DP2 is not fulfilled if the GDM-MOO method is tested with benchmark problems e.g. ZDT or DTLZ problems as in [42] or there is no testing performed for the designed method as in [78]. Otherwise, DP2 is considered to be fulfilled. The papers fulfilling DP2 include real-world case studies as in [30], real-world inspired problems that are utilized to test the developed method as in [64] or examining data from older case studies to build the MOP as in [33]. The important distinction is whether the objective functions and decision variables have real meaning for the DMs.

In Table 6, we can see that there are 17 papers that have real DMs (DP1). From these 17 papers, 16 papers also use a real-world problem to test the developed method or to solve a real-world problem (DP2). The remaining paper is [64], where the authors tested the developed method using a real-world inspired shop scheduling problem, but the authors played the roles of DMs. It can be argued that the objective functions did not have real meaning for the DMs (the authors). In total, there are 24 papers with real-world problems or real-world inspired problems.

Next, let us discuss the papers that fulfill both DP1 and DP2 according to Table 6. There were different types of DMs involved in solving various real-world MOPs. The authors in [1, 2, 30, 56, 60, 65, 66] considered various real-world case studies formed as MOPs. The papers had several different types of DMs or groups of DMs. For example, in [65], the DMs represented different companies or interest groups, e.g. farmers of the area. The DMs in [5, 50] were real humanitarian experts solving a disaster response MOP, that utilized disaster data from an earthquake. In [85], the authors considered a wine harvesting scheduling problem with an oenologist and a field manager as DMs.

The authors acted as DMs in real-world inspired problems considering portfolio optimization in [6, 27, 28, 29] and in benchmark problems in [48, 58, 72, 76, 77, 80, 89]. The validity of these GDM-MOO methods may be under question due to the following reasoning. Firstly, the authors acted as the DMs, therefore it can be argued that the objective functions do not mean anything to the DMs. In the methods using test problems and the authors acting as the DMs, this is emphasized even more. In addition, some authors used specifically selected test problems. In [48, 80], the authors used test problems that are guaranteed to find collective solutions, when the DMs give their preferences in a certain way. Since this cannot be assumed to be the case in real-world problems, how would the developed methods work in reality?

The papers [76, 77] utilizing agents representing the DMs raises some questions. In the method, the DMs are represented by agents in the evolutionary population. The DMs are supposed to give their preferences as weights for objectives to an agent. However, the evolutionary method had hundreds of agents (varying between 300 and 400), and hence the preferences of the agents were generated randomly. It is unclear how the method should work with real DMs. Are there hundreds of DMs? How the preferences of a few DMs would be incorporated into the method?

Here, we have discussed several aspects of why DP1 and DP2 should be considered when proposing new methods in the literature. Testing the methods with real-world problems will bring validity to the method, especially when combined with real DMs. The need to consider properties such as DP1 and DP2 has also been noted in the literature, for example, discussed as future research in [73, 77, 80].

Discussion of DP3

DP3 requires that the authors report the group structure, at least containing the aspects discussed in Section 3.1. Briefly, to summarize, it must be determined whether the DMs can communicate with each other or not, is the preference information shared and whether the degree of importance of the DMs is equal or not. These aspects have been captured in DP3 in Table 6.

First, let us explain how to decipher DP3 in Table 6. The column for DP3 consists of three parts of information, separated by a comma. In all of the parts, a hyphen indicates that the authors did not give this information. The first part refers to communication. In the cases, where the DMs can communicate with each other in any way, be it in-person or online, communication is allowed (marked as C_a in the table). When communication is disallowed, C_d is marked. The second part refers to the information about the preferences of the other DMs being shared with individual DMs. If preference information is shared, it is indicated by I_y and I_n is used if the information is not shared. The third part refers to the degree of importance of the DMs. In the cases, where the DMs are equal in degree of importance, we write DI_y. Otherwise, DI_n is written. This includes the cases, where e.g. the DMs are weighted differently or there is an SDM present. Let us, for example, consider the following example $-, I_n, DI_n$. Here, the first part, $-$, means that the authors did not give information about the communication among the DMs. The second part, I_n means that the preferences of the DMs were not shared with other DMs and the third part, DI_n, means that the DMs had different degrees of importance.

By observing Table 6 regarding DP3, we can see that there are various combinations of these parts. All the methods reported the degree of the importance of the DMs in some way or another. Often, the degree of importance of DMs is modeled by the weighting of the DMs as in [57, 58, 89]. Most times the weights are decided by the authors or by an SDM. However, the weighting of the DMs directly affects the solution process and the final solution selected and it is not clear how to set these weights. For example in [50], the method allows weighting of the DMs, but in testing the method the authors only consider equally weighted DMs because it was too complex of an issue to determine the importance of individual DMs without inducing bias.

In seven papers, there is an SDM or a moderator playing an important role in the solution process. The SDM was often in charge of weighting the DMs. In some cases, the SDM is a natural part of the group e.g. the manager as in [11] and a project manager in [17], who selected the DMs to participate in the solution process, weighted the DMs and also selected the final solution. However, in [6, 27, 28, 29], the methods require the SDM (or a third-party moderator) to monitor the consensus reaching process and guide the DMs during the solution process. The details of the consensus reaching process are not mentioned by the authors, except in [28], where the authors describe a general approach including the DMs discussing and applying some unspecified group techniques to reach a consensus. It may be hard to have this type of a third party available. Moreover, the person guiding the DMs in the consensus reaching process may have an effect on the final solution. This can be argued to be a desirable feature when an SDM is involved, but it is undesirable when a third-party moderator (who is assumed to be impartial) is involved. On the other hand, a competent SDM in charge of the consensus reaching process may lead to more satisfied DMs at the end of the process [34].

The communication between the DMs and sharing of preference information are not reported as often in the reviewed literature. The information presented in Table 6 regarding communication and sharing of preference information, is deduced mostly from few sentences that mention the issue in the papers. Generally, in the reviewed literature, there is very little focus on details of the group structure. In 14 papers communication is mentioned: in 8 of the papers the communication is allowed and in 6 it is disallowed. For example, in [9, 10] the DMs' can communicate through the agent-based negotiation framework, but in the other agent-based framework [76, 77] the DMs are not allowed to communicate. Regarding the sharing of the preference information, in 13 papers the authors indicate whether the preference information is shared (6 papers) or not (7 papers). In [9, 10], the DMs can see the preferences of the other DMs and the collective preference, but for example in [72, 73] the DMs do not see the preferences of the others. Most of the papers reporting whether the DMs can communicate also report whether the preference information was shared. However, for example in [48, 80], the authors mention that every DM can see the other DMs' preferences, but the communication between the DMs is not mentioned at all.

In the reviewed literature, sometimes the authors defined that their method was developed for committees [42, 87] or teams [7]. What exactly is a committee or a team in these cases, was not elaborated or defined. This type of definition is important as the methods are developed for this specific type of a group in mind. The authors in [6, 27, 28, 29], considered this issue. They provide a clear characterization of group structures originally defined by [51]. In the following, we introduce this characterization of the group structures and discuss the main properties of these group structures.

A characterization of group structures

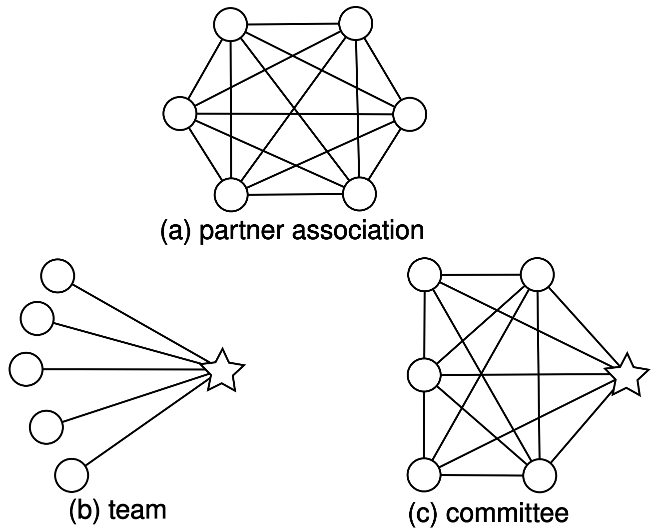

Figure 5. Group structures introduced in [51]. a) partner association b) team c) committee.

According to [51], the group structures for GDM-MOP are

(a) partner association,
(b) team and
(c) committee.

In partner association (see Figure 5 a), the whole group is responsible for the decision. There is a symmetry between the DMs indicating that all the DMs hold the same accountability of the decision. The symmetry is present in the communications and in the sharing of information. The DMs can communicate among each other and can see the preferences of the other DMs. However, the symmetry does not imply that all DMs are equally skilled negotiators or have the same system of values. The final solution is selected according to a previously agreed group decision method [26, 27].

The symmetry is missing from the other group structures called team and committee. There is an SDM in charge who chooses the group decision method to use and selects the final solution based on the preferences of the DMs. The communication and flow of information are asymmetrical. In a team, the group members only interact with the SDM (see Figure 5 b). In a committee, there is complete interaction between the group members, but the SDM still has the final say [51] (see Figure 5 c).

In addition, in all group structures, there can be a moderator or a facilitator to help in different processes [51], e.g. helping to prevent deadlocks in negotiations, helping DMs use the method or guiding the consensus reaching process. Finally, all these group structures allow allocating different degrees of importance to DMs.

Discussion of DP4

The GDM-MOO method fulfills DP4 if the preference information the DMs express is in an understandable format to the DMs. This is obviously subjective, and for example, some DMs may understand weights for objectives very well while other DMs prefer using pairwise comparisons of the solutions. Some methods allow few options for the DMs to express their preferences, but only in [6] the group does not need to agree to use the same type of preference information. However, we discuss what preference information types have been used in the different methods in the reviewed literature. In addition, we consider the advantages and disadvantages of the most common preference information types and answer the first research question.

The number of preference information types utilized in the different methods is shown in Figure 6. The preference information that is expressed in either a precise or an imprecise format, has been aggregated into the same bar. The same aggregation has also been done for weights, ranking and pairwise comparisons, regarding whether the information to be evaluated is about the objective functions or the solutions. For example, the bar for ranking contains both the ranking of the objective functions and the ranking of the solutions.

A reference point was the most popular preference information type utilized in a priori methods. The reference point had to be found by each of the DM by solving their own MOP in [6, 27, 29]. The advantage of the reference points is that they contain the aspiration levels

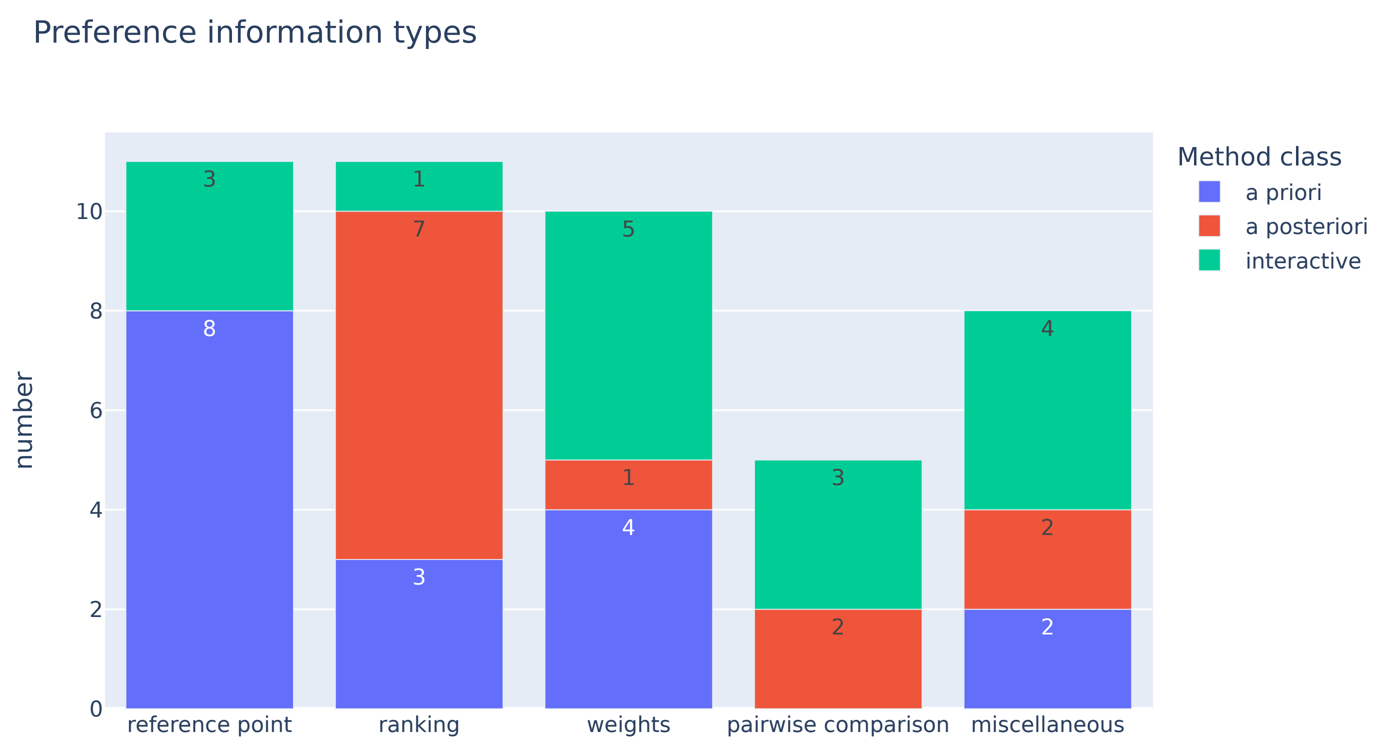

Figure 6. The preference information types in the method classes.

that the DM would like to achieve. Because of this, it can be argued that they are simple to understand for the DM. On the other hand, if the DM cannot define such aspiration levels, a reference point cannot be formed.

The next two most popular preference information types were ranking and weights. The ranking of solutions performed by voting-based social choice methods in e.g. [1, 30] was very popular in a posteriori method class, and an interactive method [46] utilized ranking of solutions. The few a priori methods [11, 17, 50] used ranking of the objective functions. The only a posteriori method using weights for objective functions was in [33]. The weights were used also in a priori and interactive methods, most often as weights for objectives as in [47, 77]. As mentioned in Chapter 2, weights are often seen as simple to understand and implement but it is hard for the DMs to provide accurate weight vectors. Additionally, the weighted sum method has issues if some of the objective functions are correlated and cannot find all Pareto optimal solution unless the problem is convex.

Pairwise comparisons were used commonly. The pairwise comparisons of the objective functions was utilized in a posteriori methods [5, 33], and the pairwise comparisons of solutions were utilized in interactive methods [18, 19, 42]. Most of the miscellaneous preference information types were one-offs and hence were aggregated into the last bar in Figure 6. These included veto, majority, credibility thresholds for an outranking model in [6, 27, 29], and classification of the objective functions in [7].

While it has been claimed that preference information types like pairwise comparison and (partial) ranking of solutions may be less cognitively demanding to provide than e.g. weights for objectives [41, 42], many methods used the less demanding preference information in combination with a larger amount of different (and more complex) preference information types. For example, in [28] the DMs are expected to give a reference point, weights for objectives and fuzzy preference relations. Similar kind of outranking-based model was used also in [6, 27, 29], where the DMs had to express various different parameters and thresholds such as majority threshold. The number of parameters required from the DMs has been noted as an issue with outranking-based methods [20]. Moreover, the authors in [28] noted that preference information they utilized can be cognitively demanding for the DMs to provide. Another example of complex preference information for the DMs to express was in [57], where the DMs express their preferences as a membership function. This type of preference information seems very hard to provide for the DM, who is the expert in the problem domain, but not in the optimization methods or preference modeling.

Many of the methods using pairwise comparisons or utility functions e.g. [30, 42, 62, 87] transform the DMs' preference information into utility functions. This approach assumes that the DMs have this kind of an underlying utility function [43] that can be inferred in some way e.g. by observing how the DM makes decisions. However, often people make decisions based on simple heuristics [43].

Here, we have discussed the precise and imprecise preference information as one entity. However, the DMs may have their own perception of the objective values and the uncertainty in the problem and according to [6, 27, 29] including imprecision in the preferences is important to consider in solving GDM-MOPs. The DMs expressed imprecise preference information, modeled using either interval or fuzzy numbers, in [6, 27, 28, 29, 50, 57, 58, 89]. However, by adding another layer of uncertainty in the form of imprecise preference information, the GDM-MOP becomes even more complex. The method must also handle the different interactions with the different layers of uncertainty e.g. from the preferences and the uncertainty related to the problem e.g. simulation-optimization model as in [30].

In summary, the relevant issue regarding DP4 is balancing between the understandability of the preference information for human DMs and using preference information types that provide some desired properties in the preference model.

Discussion of DP5 and DP6

Next, we consider DP5, DP6. In addition, we summarize the answers to research questions 2 and 3 as they are linked to DP5 and DP6. DP5 aims to ensure that the preferences of the DMs are taken into account in the method. This means that the DMs' preferences are incorporated into the method in some manner. Figure 7 shows how many of the methods used indirect and direct approaches. DP6 aims to ensure that the GDM-MOO method supports the DMs in selecting the final solution. The final solution must be selected to solve the GDM-MOP successfully. Figure 8 shows how many of the methods selected the final solution.

In the following, we discuss Figures 7, 8 and also refer to Tables 3 and 5. According to Figure 7, indirect and direct approaches were both used in the reviewed literature. As discussed in Section 3.2, if there are multiple collective solutions found, a group decision method must be used to select the final solution.

Of the total of 40 papers, in 22 papers the proposed method provided support for selecting the final solution. So, in almost half of the GDM-MOO methods, the final solution was not selected and the GDM-MOP was not fully solved. As nine out of ten a posteriori GDM-MOO methods selected the final solution, only 13 papers select the final solution of the total of 30 papers in a priori and interactive methods. As far as a priori methods are concerned, in five of 17 papers, the final solution was selected. Considering interactive methods, in eight of the total of 13 papers, the final solution was selected. It is clear that this is not an ideal situation for the GDM-MOO literature for real-world relevancy.

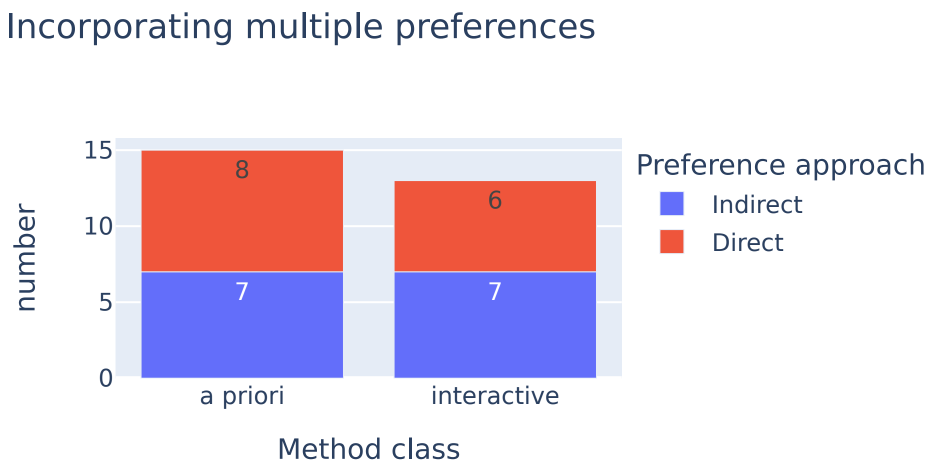

Figure 7. The preference approaches in a priori and interactive methods

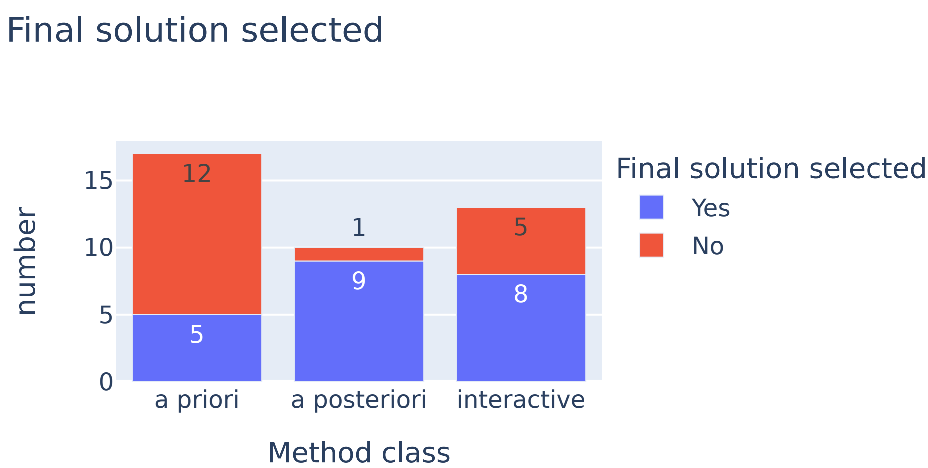

Figure 8. The number of papers selecting the final solution.

However, the reviewed literature sheds some light on why this situation has occurred. The final solution selection phase is often dismissed in the reviewed GDM-MOO literature. This is manifested in mainly two ways. Firstly, the authors in [9, 10, 42, 77, 89] explicitly mention that the method aims to find a set of collective solutions, instead of a final solution. Secondly, the final solution selection was seen as a trivial part of the process that can be solved by utilizing any already existing group decision method. However, many times the suggested group decision method was not applied and the final solution was not selected. For example, in [31], the authors state that the DMs select the final solution by negotiations and in [77], the authors suggest that the DMs may use some criterion as the group decision method e.g. select the best non-dominated solution, to select the final solution. The authors in [77] did not specify what the best non-dominated solution in this context is.

However, the reviewed literature did contain GDM-MOO methods utilizing either indirect or direct approaches and selecting the final solution. For example, in [11], the method utilized a direct approach and an SDM selected the final solution. In an interactive method in [85], the DMs were gradually shown generated Pareto optimal solutions, until a solution was found which both of the DMs determined as acceptable. The GDM-MOO methods utilizing an indirect approach selected the final solution most times, by using different group decision methods e.g. utilizing a consensus reaching process to select the final solution in [46], or finding the final solution using the weighted sum method with a collective weight vector in [50] or using the Borda count method to select the final solution in [6].

In [27, 29], the authors brought up the case that much of the interactive GDM-MOO methods assume that a collective preference is transitive, which as discussed in 3.2 is not the case. Therefore in using indirect approach this should be considered and not assuming that the transitivity property is in collective preference and hence a solution should be acceptable to the DM.

Next, we discuss a posteriori methods in light of DP6. As mentioned, 9 out of 10 a posteriori methods selected the final solution. The methods utilized several different social choice methods, and the authors indicated which collective solution was determined as the final solution. The only exception was in [56], where various social choice methods were used to find several collective solutions, but the authors did not clarify which was selected as the final one.

However, there is a concern regarding some of the reviewed a posteriori methods. As mentioned, the DMs in [1, 2, 30, 56, 60, 65, 66] expressed their preferences in the problem formulation stage, and then the DM's preferences were modeled with utility functions. Then, a set of non-dominated solutions was generated with an NSGA-II. Next, the DMs were expected to be able to rank a large number (a few dozens) of non-dominated solutions using different social choice methods in [1, 2, 30, 56, 60, 65, 66]. For example, in [30], the DMs had to rank 35 solutions in five different uncertainty scenarios.

However, ranking all of the found non-dominated solutions would not be possible if there are hundreds or thousands of non-dominated solutions, which is not a rare occurrence when using EMOAs. Therefore, some sort of filtering must be done to select a (reasonably sized) subset of solutions to show to the DMs. There was no discussion about this issue in the reviewed literature and only in [33], the DMs have the option to indicate how many solutions they are prepared to rank. The above discussion raises the question of whether the DMs are ranking the non-dominated solutions or their utility functions are performing the ranking of solutions and hence the DMs are not actively involved into the solution process. For example, in [65], the authors mention that the DMs' utility functions are used to rank the solutions.

Discussion of DP7

DP7 brings up the acceptance of the final solution among the DMs. As discussed in Section 3.3, the final solution is the solution most preferred by the group, but in many cases, the individual DMs may not agree on the final solution. The acceptance of the final solution is especially relevant in real-world MOPs where the DMs may not implement the solution they do not agree with. Hence, the acceptance of the final solution must be considered in the method.

Let us discuss Table 6. In the table, the last column indicates, whether DP7 was considered in the paper with symbol ✓or not with symbol ✗. The reasoning is the following. First, the final solution must be selected and additionally, there must be some validation of the acceptance among the individual DMs. The validation means using different acceptance measures such as consensus measures or plainly asking the DMs questions, such as do you accept the final solution or what do you think of the final solution. In the following, we discuss the types of validation performed in the papers. If this type of validation is not reported by the authors, in this review, we consider that D7 is not fulfilled.

A popular way of validating the acceptance of the final solution is to rely on a group decision method that the group has agreed to use. For example in [1, 2, 30, 60, 65, 66] the DMs have agreed to use social choice methods to select the final solution. The final solution selected with the social choice method is a socially optimal solution and accepted by the DMs. Another approach is using consensus measures. In the reviewed literature, several different consensus measures were used to determine the acceptance degree of the final solution e.g. in [6, 27, 28, 62]. When the SDM selects the final solution as in [11, 17], the acceptance of the other DMs is not relevant, since the SDM's acceptance is the only thing that can prevent the solution from being implemented. Although, in [27], it was supposed that the SDM acts as an altruistic dictator and in some way considers the acceptance of the final solution among all the DMs. These types of approaches rely on the DMs respecting the group decision method selected earlier in the decision making. However, it is not clear how the DMs select the group decision method and if the DMs do not select it, how can it be assumed that the DMs accept the final solution?

Another way of bringing up some validation of the acceptance of the final solution was asking the DMs what they felt about the final solution. In [46], where the majority of the DMs reported that the group interaction played a great role in the GDM process and it positively contributed to their comprehension of the problem considered. Furthermore, the DMs showed a high degree of confidence in the final solution. Also in [33], the authors mention that DMs were asked to comment on the final solution regarding their preference information. Another way was presented in [5, 11], where the authors discussed what solutions individual DMs would have preferred and why the final solution was selected.

In summary, there are different ways to take into account the acceptance of the final solution among the DMs. In the reviewed literature, the most often used way was relying on the group decision method to guarantee acceptance based on the fact that the DMs agreed to use that specific group decision method. Only a few papers asked for the DMs' feedback on the acceptance of the final solution.

5.3 Limitations of the conducted work and future research directions

In this section, we discuss the limitations of the literature review. Furthermore, we discuss several important future research directions found in the reviewed literature. Firstly, the searches were conducted in July of 2022, but the following steps of including, excluding and reading the papers, analyzing the results and writing the thesis have taken almost a year. This means that the searches may be somewhat outdated. In addition, the amount of knowledge gained by preparing this review would let us form better search queries to possibly find something new. The focus on GDM-MOO is vague and several different approaches in GDM have not been considered in this review, the biggest example being game theoretical approaches. However, as a Master's thesis, it is not possible to include everything.

Furthermore, the conducted classifications could be improved. The preference incorporation could be split in more detail than to indirect and direct approaches. This could include e.g. considering different group decision method types in the classification. In addition, the classification of GDM-MOO methods as a priori, a posteriori and interactive ones is not the only one that can be used. Another possible way would be to consider in the classification of how the GDM-MOP is solved, e.g. in an asynchronous manner, where the MOP is solved first and then the GDM problem and methods where the MOP and GDM problem are solved synchronously. This type of classification could emphasize the role of GDM in MOO better.

Overall, solving GDM-MOPs requires more research. Future research directions include developing GDM-MOO methods for different groups of DMs. We discussed group structures to consider the different types of groups but these group structures (or something like them) are not used in the literature. As mentioned, only in [6, 27, 28, 29], the group structure was defined and e.g. in [87] is was mentioned that the method is designed for committees, however, there was no further information about the group structure. In addition, a DM may give preferences in a manipulative manner e.g. by giving more extreme preferences than they actually prefer to move the collective preference more towards their preference. In the reviewed literature, the manipulative behavior of the DMs was only considered in [9], and there are multiple issues to consider when trying to detect and prevent it. For example, consider an interactive method. What if the DM learns from the problem and gives suddenly different preferences? How to detect if the change is due to learning or from a manipulative behavior of the DM? Furthermore, if the manipulative behavior is detected, how should it be handled? Should the DM be punished e.g. by lowering their degree of importance? There are no simple answers, and the unique combination of GDM-MOO makes the issue even more difficult to handle.

There are still more issues to consider, for example, how to decide the degree of importance of the DMs, how to guarantee the acceptance of the final solution in situations where the DMs are independent and can decide not to implement the final solution. The DMs should be more actively involved in the solution process and feedback should be asked from the DMs, especially regarding the acceptance of the final solution. The latter was only considered in few papers e.g. in [46]. Other issues relate to the methods used and the communication of the DMs, the interactions among the DMs and using real-world problems. As 70% of the methods (used in the reviewed literature) were either based on the weighted sum or NSGA-II, other types of MOO methods should also be considered in solving GDM-MOPs. Furthermore, as mentioned, much of the literature did not use real DMs or solve real-world problems in testing the proposed methods. Some methods used a consensus reaching process and group discussions to promote consensus among the DMs. However, details of these approaches were not mentioned. Future research should fill these gaps in the literature.

Other types of future research directions include considering solving GDM-MOPs with more than 20 DMs, referred to as large-scale GDM-MOPs, which is becoming a more and more important problem domain in the expansion of the current technologies such as social networks and e-democracy [45]. Furthermore, in the reviewed literature, there are no GDM-MOO methods directed at cases, where the DMs have different sets of objective functions. In addition, handling imprecise information and considering different roles of the DMs is seen as an important research direction in GDM-MOO [27, 29].

6 Conclusions

In this thesis, we concentrated on assessing the current state of the literature regarding GDM and MOPs. The main concepts of MOO and GDM were introduced and common terminology was laid for GDM-MOO. We explained how the systematic literature was conducted. Then the papers included were discussed in some detail. Following, we summarized what we found in the 40 papers that were included in this review.

The papers with GDM-MOO methods were classified according to the role of the DM in the solution process to a priori, a posteriori and interactive methods. We considered the preference information types the DMs expressed and two different ways of handling multiple preferences from several DMs, called indirect and direct approaches. In addition, we investigated the phase of the final solution selection in the papers. Lastly, we proposed desirable properties for GDM-MOO methods. The desirable properties were the result of the work done, including important aspects to consider in solving GDM-MOPs. We discussed the papers in the light of the desirable properties. Moreover, we suggested some ways to consider these desirable properties.

This review aimed at investigating the state-of-the-art of GDM-MOO literature and finding ways to solve these types of problems. As discussed, there are different approaches taken in the GDM-MOO literature. Overall, most of the methods are either specific to a problem area, to a group structure or to a theoretical test framework. A large portion of the literature does not consider the final solution selection. In this thesis, we also discussed why different types of group structures are meaningful and suggested a way of characterizing group structures of different types of groups.

We have detected gaps in the current GDM-MOO literature such as considering manipulative behavior, large-scale GDM-MOPs, using other types of MOO methods than the weighted sum or NSGA-II and the lack of using real DMs and real-world problems in testing the proposed methods. The weaknesses detected in the literature include not selecting the final solution, not considering what type of groups the method has been developed for and not asking for any feedback from the DMs on the acceptance of the final solution.

In summary, we reviewed what has been done in solving GDM-MOPs. We answered the research questions of how to solve GDM-MOPs, what type of preference information was used, how the DMs' preferences were incorporated into the solution process and how the final solution was selected. Furthermore, we suggested some desirable properties. In future research, these properties should be tested and further enhanced with the knowledge gained from experiments involving real DMs and real-world GDM-MOPs. The DMs should be inquired about the relevant aspects of the designed methods, e.g. whether the DMs accept the final solution or think that the method is understandable and whether it works well.

Acknowledgments

The author would like to deeply thank his supervisors Kaisa Miettinen and Johanna Silvennoinen for their guidance and feedback with respect to the writing of this thesis.

This thesis is related to the thematic profiling area of Decision analytics utilizing causal models and multiobjective optimization (DEMO) of University of Jyväskylä, and has been partly funded by the Academy of Finland (UTOPIA project number: 352784).

Bibliography

[1] M. R. Alizadeh, M. R. Nikoo, and G. R. Rakhshandehroo. “Developing a multi-objective conflict-resolution model for optimal groundwater management based on fallback bargaining models and social choice rules: A case study”. In: *Water Resources Management* 31.5 (2017), pp. 1457–1472.

[2] M. R. Alizadeh, M. R. Nikoo, and G. R. Rakhshandehroo. “Hydro-environmental management of groundwater resources: A fuzzy-based multi-objective compromise approach”. In: *Journal of Hydrology* 551 (2017), pp. 540–554.

[3] K. J. Arrow. *Social Choice and Individual Values*. 3rd ed. Yale University Press, 2012.

[4] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996.

[5] H. Baharmand, T. Comes, and M. Lauras. “Supporting group decision makers to locate temporary relief distribution centres after sudden-onset disasters: A case study of the 2015 Nepal earthquake”. In: *International Journal of Disaster Risk Reduction* 45 (2020), article 101455.

[6] F. Balderas, E. Fernández, L. Cruz-Reyes, C. Gómez-Santillán, and N. Rangel-Valdez. “Solving group multi-objective optimization problems by optimizing consensus through multi-criteria ordinal classification”. In: *European Journal of Operational Research* 297.3 (2022), pp. 1014–1029.

[7] C. Baril, S. Yacout, and B. Clément. “An interactive multi-objective algorithm for decentralized decision making in product design”. In: *Optimization and Engineering* 13.1 (2012), pp. 121–150.

[8] G. W. Bassett Jr and J. Persky. “Robust voting”. In: *Public Choice* 99.3-4 (1999), pp. 299–310.

[9] S. Bechikh, L. B. Said, and K. Ghédira. “Group preference-based evolutionary multi-objective optimization with non-equally important decision makers: Application to the portfolio selection problem”. In: *International Journal of Computer Information Systems and Industrial Management Applications* 5.1 (2013), pp. 278–288.

[10] S. Bechikh, L. B. Said, and K. Ghédira. “Negotiating decision makers’ reference points for group preference-based evolutionary multi-objective optimization”. In: *2011 11th International Conference on Hybrid Intelligent Systems (HIS)*. IEEE. 2011, pp. 377–382.

[11] D. Borissova and I. Mustakerov. “A two-stage placement algorithm with multi-objective optimization and group decision making”. In: *Cybernetics and Information Technologies* 17.1 (2017), pp. 87–103.

[12] S. J. Brams and D. M. Kilgour. “Fallback bargaining”. In: *Group Decision and Negotiation* 10 (2001), pp. 287–316.

[13] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz. “Learning value functions in interactive evolutionary multiobjective optimization”. In: *IEEE Transactions on Evolutionary Computation* 19.1 (2014), pp. 88–102.

[14] F. J. Cabrerizo, J. M. Moreno, I. J. Pérez, and E. Herrera-Viedma. “Analyzing consensus approaches in fuzzy group decision making: Advantages and drawbacks”. In: *Soft Computing* 14.5 (2010), pp. 451–463.

[15] X. Chao, Y. Dong, G. Kou, and Y. Peng. “How to determine the consensus threshold in group decision making: a method based on efficiency benchmark using benefit and cost insight”. In: *Annals of Operations Research* 316.1 (2022), pp. 143–177.

[16] S-J. Chen and C-L. Hwang. *Fuzzy Multiple Attribute Decision Making Methods*. Springer, 1992.

[17] W.-Y. Chiu, S. H. Manoharan, and T.-Y. Huang. “Weight induced norm approach to group decision making for multiobjective optimization problems in systems engineering”. In: *IEEE Systems Journal* 14.2 (2019), pp. 1580–1591.

[18] D. Cinalli, L. Martı, N. Sanchez-Pi, and A. C. B. Garcia. “Extending collective intelligence evolutionary algorithms: A facility location problem application”. In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2020, pp. 1–8.

[19] D. Cinalli, L. Martı, N. Sanchez-Pi, and A. C. B. Garcia. “Integrating collective intelligence into evolutionary multi-objective algorithms: Interactive preferences”. In:

2015 Latin America Congress on Computational Intelligence (LA-CCI). IEEE. 2015, pp. 1–6.

[20] C. A. C. Coello. “Handling preferences in evolutionary multiobjective optimization: A survey”. In: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00*. IEEE. 2000, pp. 30–37.

[21] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd ed. Springer, 2007.

[22] K. Deb. “Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction”. In: *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*. Ed. by L. Wang, A. Ng, and K. Deb. Springer, 2011, pp. 3–34.

[23] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197.

[25] M. Ehrgott. *Multicriteria Optimization*. Springer, 2005.

[26] E. Fernández, S. Bernal, J. Navarro, and R. Olmedo. “An outranking-based fuzzy logic model for collaborative group preferences”. In: *TOP* 18.2 (2010), pp. 444–464.

[27] E. Fernández, C. Gómez-Santillán, N. Rangel-Valdez, and L. Cruz-Reyes. “Group multi-objective optimization under imprecision and uncertainty using a novel interval outranking approach”. In: *Group Decision and Negotiation* (2022), pp. 1–50.

[28] E. Fernández and R. Olmedo. “An outranking-based general approach to solving group multi-objective optimization problems”. In: *European Journal of Operational Research* 225.3 (2013), pp. 497–506.

[29] E. Fernández, N. Rangel-Valdez, L. Cruz-Reyes, and C. Gomez-Santillan. “A new approach to group multi-objective optimization under imperfect information and its application to project portfolio optimization”. In: *Applied Sciences* 11.10 (2021), article 4575.

[30] M. Ghorbani Mooselu, M. R. Nikoo, and M. Sadegh. “A fuzzy multi-stakeholder socio-optimal model for water and waste load allocation”. In: *Environmental Monitoring and Assessment* 191.6 (2019), pp. 1–16.

[31] P. Guarneri and M. M. Wiecek. “Pareto-based negotiation in distributed multidisciplinary design”. In: *Structural and Multidisciplinary Optimization* 53.4 (2016), pp. 657–671.

[32] T. Hacardiaux, C. Defryn, J.-S. Tancrez, and L. Verdonck. “Balancing partner preferences for logistics costs and carbon footprint in a horizontal cooperation”. In: *OR Spectrum* 44.1 (2022), pp. 121–153.

[33] Y. Hadas and O. E. Nahum. “Urban bus network of priority lanes: A combined multi-objective, multi-criteria and group decision-making approach”. In: *Transport Policy* 52 (2016), pp. 186–196.

[34] E. Herrera-Viedma, F. J. Cabrerizo, J. Kacprzyk, and W. Pedrycz. “A review of soft consensus models in a fuzzy environment”. In: *Information Fusion* 17 (2014), pp. 4–13.

[35] E. Herrera-Viedma, F. Herrera, and F. Chiclana. “A consensus model for multiperson decision making with different preference structures”. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 32.3 (2002), pp. 394–402.

[36] M. Horowitz, B. M. Stewart, D. Tingley, M. Bishop, L. Resnick Samotin, M. Roberts, W. Chang, B. Mellers, and P. Tetlock. “What makes foreign policy teams tick: Explaining variation in group performance at geopolitical forecasting”. In: *The Journal of Politics* 81.4 (2019), pp. 1388–1404.

[37] C-L. Hwang and M. Lin. *Group Decision Making under Multiple Criteria: Methods and Applications*. Springer, 1987.

[38] C-L. Hwang and K. Yoon. *Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey*. 1981.

[39] Y. Jin. “A comprehensive survey of fitness approximation in evolutionary computation”. In: *Soft computing* 9.1 (2005), pp. 3–12.

[40] J. Kacprzyk. “Group decision making with a fuzzy linguistic majority”. In: *Fuzzy Sets and Systems* 18.2 (1986), pp. 105–118.

[41] M. Kadziński and T. Tervonen. “Robust multi-criteria ranking with additive value models and holistic pair-wise preference statements”. In: *European Journal of Operational Research* 228.1 (2013), pp. 169–180.

[42] M. Kadziński and M. K. Tomczyk. “Interactive evolutionary multiple objective optimization for group decision incorporating value-based preference disaggregation methods”. In: *Group Decision and Negotiation* 26.6 (2017), pp. 1215–1240.

[43] R. L. Keeney, H. Raiffa, and R. F. Meyer. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Cambridge University Press, 1993.

[44] I. Y. Kim and O. L. De Weck. “Adaptive weighted-sum method for bi-objective optimization: Pareto front generation”. In: *Structural and Multidisciplinary Optimization* 29 (2005), pp. 149–158.

[45] Á. Labella, Y. Liu, R. M. Rodrıguez, and L. Martı nez. “Analyzing the performance of classical consensus models in large scale group decision making: A comparative study”. In: *Applied Soft Computing* 67 (2018), pp. 677–690.

[46] H. S. Lewis and T. W. Butler. “An interactive framework for multi-person, multiobjective decisions”. In: *Decision Sciences* 24.1 (1993), pp. 1–22.

[47] G. Liu, G. Wu, T. Zheng, and Q. Ling. “Integrating preference based weighted sum into evolutionary multi-objective optimization”. In: *2011 Seventh International Conference on Natural Computation*. IEEE. 2011, pp. 1251–1255.

[48] W. Liu, W. Luo, X. Lin, M. Li, and S. Yang. “Evolutionary approach to multiparty multiobjective optimization problems with common Pareto optimal solutions”. In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2020, pp. 1–9.

[49] J. Lu and D. Ruan. *Multi-Objective Group Decision Making: Methods, Software and Applications with Fuzzy Set Techniques*. Imperial College Press, 2007.

[50] R. Maharjan and S. Hanaoka. “A multi-actor multi-objective optimization approach for locating temporary logistics hubs during disaster response”. In: *Journal of Humanitarian Logistics and Supply Chain Management* 8.1 (2018), pp. 2–21.

[51] G. M. Marakas. *Decision Support Systems in the 21st Century*. Prentice Hall, 2003.

[52] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[53] K. Miettinen, J. Hakanen, and D. Podkopaev. "Interactive Nonlinear Multiobjective Optimization Methods". In: *Multiple Criteria Decision Analysis: State of the Art Surveys*. Ed. by S. Greco, M. Ehrgott, and J. R. Figueira. Springer, 2016, pp. 927–976.

[54] K. Miettinen and M. M Mäkelä. "On scalarizing functions in multiobjective optimization". In: *OR Spectrum* 24 (2002), pp. 193–213.

[55] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. "Introduction to Multiobjective Optimization: Interactive Approaches". In: *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Ed. by J. Branke, K. Deb, K. Miettinen, and R. Słowiński. Springer, 2008, pp. 27–57.

[56] A. R. Nafarzadegan, H. Vagharfard, M. R. Nikoo, and A. Nohegar. "Socially-optimal and Nash Pareto-based alternatives for water allocation under uncertainty: An approach and application". In: *Water Resources Management* 32.9 (2018), pp. 2985–3000.

[57] K. Nag, T. Pal, R. K. Mudi, and N. R. Pal. "Robust multiobjective optimization with robust consensus". In: *IEEE Transactions on Fuzzy Systems* 26.6 (2018), pp. 3743–3754.

[58] K. Nag, T. Pal, and N. R. Pal. "Robust consensus: A new measure for multicriteria robust group decision making problems using evolutionary approach". In: *International Conference on Artificial Intelligence and Soft Computing*. Springer. 2014, pp. 384–394.

[59] H. Nakayama. "Aspiration level approach to interactive multi-objective programming and its applications". In: *Advances in multicriteria analysis* (1995), pp. 147–174.

[60] S. S. Naserizade, M. R. Nikoo, H. Montaseri, and M. R. Alizadeh. "A hybrid fuzzy-probabilistic bargaining approach for multi-objective optimization of contamination warning sensors in water distribution systems". In: *Group Decision and Negotiation* 30.3 (2021), pp. 641–663.

[61] C. Okoli. “A guide to conducting a standalone systematic literature review”. In: *Communications of the Association for Information Systems* 37.1 (2015), article 43.

[62] S. L. C. Oliveira and P. A. V. Ferreira. “Bi-objective optimisation with multiple decision-makers: A convex approach to attain majority solutions”. In: *Journal of the Operational Research Society* 51.3 (2000), pp. 333–340.

[63] I. Palomares, F. J. Estrella, L. Martınez, and F. Herrera. “Consensus under a fuzzy context: Taxonomy, analysis framework AFRYCA and experimental case of study”. In: *Information Fusion* 20 (2014), pp. 252–271.

[64] J. Pfeiffer, U. Golle, and F. Rothlauf. “Reference point based multi-objective evolutionary algorithms for group decisions”. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. 2008, pp. 697–704.

[65] E. Raei, M. R. Alizadeh, M. R. Nikoo, and J. Adamowski. “Multi-objective decision-making for green infrastructure planning (LID-BMPs) in urban storm water management under uncertainty”. In: *Journal of Hydrology* 579 (2019), article 124091.

[66] E. Raei, M. R. Nikoo, and S. Pourshahabi. “A multi-objective simulation-optimization model for in situ bioremediation of groundwater contamination: Application of bargaining theory”. In: *Journal of Hydrology* 551 (2017), pp. 407–422.

[67] H. Raiffa, J. Richardson, and D. Metcalfe. *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Harvard University Press, 2002.

[68] D. Richards. “Is strategic decision making chaotic?” In: *Behavioral Science* 35.3 (1990), pp. 219–232.

[69] A. Rothstein and C.T. Butler. *On Conflict Consensus: A Handbook on Formal Consensus Decisionmaking*. Portland: Food Not Bombs Publishing, 1987.

[70] B. Roy and V. Mousseau. “A theoretical framework for analysing the notion of relative importance of criteria”. In: *Journal of Multi-Criteria Decision Analysis* 5.2 (1996), pp. 145–159.

[71] Bernard Roy. *Multicriteria Methodology for Decision Aiding*. Springer, 1996.

[72] B. Rubenstein-Montano and R. Malaga. “A co-evolutionary approach to strategy design for decision makers in complex negotiation situations”. In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. IEEE. 2000, pp. 1–9.

[73] B. Rubenstein-Montano and R. A. Malaga. “A weighted sum genetic algorithm to support multiple-party multiple-objective negotiations”. In: *IEEE Transactions on Evolutionary Computation* 6.4 (2002), pp. 366–377.

[74] F. Ruiz, M. Luque, and K. Miettinen. “Improving the computational efficiency in a global formulation (GLIDE) for interactive multiobjective optimization”. In: *Annals of Operations Research* 197 (2012), pp. 47–70.

[75] R. W. Saaty. “The analytic hierarchy process—what it is and how it is used”. In: *Mathematical Modelling* 9.3-5 (1987), pp. 161–176.

[76] H. Sakamoto, K. Nakamoto, and K. Ohnishi. “Acquiring consensus solutions by multi-human-agent-based evolutionary computation”. In: *2021 5th IEEE International Conference on Cybernetics (CYBCONF)*. IEEE. 2021, pp. 12–18.

[77] H. Sakamoto, K. Nakamoto, and K. Ohnishi. “Evolutionary computation system solving group decision making multiobjective problems for human groups”. In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 26.2 (2022), pp. 196–205.

[78] K. Sampath, S. K. R. Danda, K. Kumar, K. Narayanam, P. Dayama, and S. Sankagiri. “Spot collaborative shipping sans orchestrator using Blockchain”. In: *2020 IEEE International Conference on Blockchain*. IEEE. 2020, pp. 371–378.

[79] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, 1985.

[80] Z. She, W. Luo, Y. Chang, X. Lin, and Y. Tan. “A new evolutionary approach to multiparty multiobjective optimization”. In: *International Conference on Swarm Intelligence*. Springer. 2021, pp. 58–69.

[81] H. S. Shih, H. J. Shyur, and E. S. Lee. “An extension of TOPSIS for group decision making”. In: *Mathematical and Computer Modelling* 45.7-8 (2007), pp. 801–813.

[82] G. Stasser and W. Titus. “Pooling of unshared information in group decision making: Biased information sampling during discussion”. In: *Journal of Personality and Social Psychology* 48.6 (1985), article 1467.

[83] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley, 1986.

[84] E. Triantaphyllou. *Multi-criteria Decision Making Methods: A Comparative Study*. Springer, 2000.

[85] M. Varas, F. Basso, S. Maturana, D. Osorio, and R. Pezoa. “A multi-objective approach for supporting wine grape harvest operations”. In: *Computers & Industrial Engineering* 145 (2020), article 106497.

[86] H. Wang, M. Olhofer, and Y. Jin. “A mini-review on preference modeling and articulation in multi-objective optimization: Current status and challenges”. In: *Complex & Intelligent Systems* 3 (2017), pp. 233–245.

[87] R. E. Wendell. “Multiple objective mathematical programming with respect to multiple decision-makers”. In: *Operations Research* 28.5 (1980), pp. 1100–1111.

[88] A. P. Wierzbicki. “A mathematical basis for satisficing decision making”. In: *Mathematical Modelling* 3.5 (1982), pp. 391–405.

[89] J. Xiong, X. Tan, K.-W. Yang, and Y.-W. Chen. “Fuzzy group decision making for multiobjective problems: Tradeoff between consensus and robustness”. In: *Journal of Applied Mathematics* 2013 (2013).

[90] L. Zadeh. “Optimality and non-scalar-valued performance criteria”. In: *IEEE transactions on Automatic Control* 8.1 (1963), pp. 59–60.

[91] A. Zahedi Khameneh and A. Kilicman. “Some construction methods of aggregation operators in decision-making problems: An overview”. In: *Symmetry* 12.5 (2020), article 694.