Check for
updates

On the dominant set selection problem and its application to value alignment

Marc Serramia[1,2] · Maite López-Sánchez[2] · Stefano Moretti[3] ·
Juan Antonio Rodríguez-Aguilar[1]

Abstract

Decision makers can often be confronted with the need to select a subset of objects from a set of candidate objects by just counting on preferences regarding the objects' features. Here we formalise this problem as the *dominant set selection problem*. Solving this problem amounts to finding the preferences over all possible sets of objects. We accomplish so by: (i) grounding the preferences over features to preferences over the objects themselves; and (ii) lifting these preferences to preferences over all possible sets of objects. This is achieved by combining lex-cel –a method from the literature—with our novel anti-lex-cel method, which we formally (and thoroughly) study. Furthermore, we provide a binary integer program encoding to solve the problem. Finally, we illustrate our overall approach by applying it to the selection of value-aligned norm systems.

1 Introduction

Some actual-world decision making problems require to select an array of elements despite decision makers only counting on preferences over the elements' features. Some examples are committee selection [13], or college admissions [9, 17]. Considering this last example, picture the following situation. A school head master must decide on which students to grant admission to. For that, the head master leverages on the admission policy of the school, which, for instance, prioritises some minorities, or fosters impoverished neighbourhoods. Such policies can be cast as preferences over the students' features. Nonetheless, the head master lacks of a straightforward manner to rank all possible sets of students, since these features somehow pose a multi-criteria problem. Moreover, there is a further dimension of complexity: some sets may not be eligible (e.g. because of limited budget,

✉ Juan Antonio Rodríguez-Aguilar
jar@iiia.csic.es

Marc Serramia
marcserr@iiia.csic.es

1 IIIA-CSIC, Campus de la UAB, 08193 Bellaterra, Spain

2 Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007 Barcelona, Spain

3 LAMSADE Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France

or unfulfilment of minority quotas). And yet, despite only counting on preferences over features and not sets, the head master must select the most preferred set of students. Interestingly, we can think of many other, similar set selection problems, such as selecting the team of players for a match (where we prefer some types of players over others), personnel selection (where some capabilities may be preferred over others), selecting regulatory norms (where we prefer norms that are more aligned with some moral values), etc. The goal of this paper is to design the tools to help decision makers select the "most preferred" set in this type of problem, which hereafter we will refer to as *dominant set selection problem* (DSSP). Dominance characterises maximal preference in a formal (and particular) way.

In more general terms, assuming that we have sets of objects representing alternatives in a decision making process, the problem that we tackle is that of finding the most preferred set of objects. This decision must be made based on preference information over the *features* characterising the objects. For instance, in our admission example, ethnic group, neighbourhood, and studied subjects constitute some possible features. Furthermore, as noticed above, when dealing with decisions, preferences are not the only aspect to consider. Thus, we also require that the selected set does comply with some *feasibility constraints*, be them structural –due to relationships between the objects–, or inherent to the application domain.

In order to solve the *dominant set selection problem*, we propose to proceed as follows: (1) extract preferences over single objects based on preferences over objects' features; (2) rank all possible sets of objects; and (3) select the most preferred and feasible set of objects. For that, we resort to recent, seminal work in the realm of decision making and social choice theory, namely social rankings [15] and its solutions [1, 6, 11, 12]. By adapting lex-cel, a ranking method introduced in [6], we are able to obtain a ranking over single objects from the feature preferences. Ultimately, our goal is to rank all sets of objects considering this element ranking, in other words, lifting the element ranking to a set ranking. This lifting procedure is very similar to the *ranking sets of objects* problem, which has been extensively studied in the social choice literature [3]. Example solutions to this problem are the maxmin and minmax [2] or leximin and leximax [16] functions. Unfortunately, this problem considers a total order of elements instead of an element ranking. Hence, for the purpose of this paper, we cannot readily use any of these approaches. Instead, here we design a novel *ranking function*, the so-called *anti-lex-cel*. This function receives as an input a ranking over single objects (obtained through lex-cel), and builds a ranking over all possible sets of these objects such that the most preferred feasible set in the ranking is the solution to the dominant set selection problem. The combination of the *lex-cel* ranking described in [6] with our novel *anti-lex-cel* ranking helps us produce our intended ranking over all possible sets of objects, and hence solve the core of the dominant set selection problem.

From a pragmatic perspective, building a ranking over all sets of objects turns out to be costly. Hence, we show how to solve the dominant set selection problem while avoiding the cost of explicitly building a whole ranking. In particular, we show how to to encode it as a binary integer program (BIP) so that it can be solved with the aid of off-the-shelf solvers. Importantly, we prove that the proposed encoding adheres to the ranking produced after *lex-cel* and *anti-lex-cel*, and that the solution to our BIP is equivalent to that of the dominant set selection problem. We illustrate the application of our method to a value-alignment problem initially introduced in [22] and subsequently investigated from a qualitative perspective in [20]. In particular, given a collection of candidate norms, we investigate the

selection of the (sub)set of norms, the so-called norm system[1], that is best *aligned* with the moral values[2] in a value system. The dominant set selection problem in this case is performed according to the following principle: the more preferred the moral values promoted by a norm system, the more preferred the norm system, or, in other words, the more dominant with respect to value alignment. Here the decision maker must consider: the preferences over moral values in the value system, the promotion relationship between norms and moral values (which can be interpreted as norm features), and the feasibility conditions based on the relationships between norms.

Notice that our approach differs from the norm selection method proposed in [22], which follows a quantitative approach despite the qualitative nature of the information available to the decision maker. Since the approach in [22] follows [4], the decision maker is forced to quantify the relations between norms and values by specifying the degrees of value promotion of norms. We argue that this is hard to ascertain and, as noted in [19], transforming qualitative information into numerical data is prone to errors and biases. In fact, this is a general claim that can be applied when solving the dominant set selection problem. Therefore, in this paper we opt for a qualitative approach with the aim of keeping the decision making process as intuitive as possible.

To summarise, in this paper we make the following contributions:

1. Formalisation of a novel qualitative decision-making problem, the so-called *dominant set selection problem* (DSSP).
2. Formalisation and study of a novel preference lifting function called *anti-lex-cel*. We provide an axiomatic characterisation of *anti-lex-cel*, and we show that it generalises former results in the social choice literature in [7].
3. Development of a novel method for solving the DSSP based on the combination of the *lex-cel* ranking function in [6] with our novel *anti-lex-cel* ranking function.
4. A binary integer program (BIP) encoding that is proven to solve the DSSP while avoiding the cost of explicitly building a whole ranking over all possible sets of objects.
5. An application of our methodology to the value-alignment problem described in [20, 22].

This work significantly extends our previous work in [20] in two main respects. First, here we present a general formalisation and solving method for the DSSP, hence going beyond [20], which solely focused on composing value-aligned norm systems, namely on a particular DSSP. Second, here we add with respect to [20] a thorough axiomatic characterisation of *anti-lex-cel*, a formal proof of its uniqueness, and results that show the generality of anti-lex-cel with respect to existing results in the social choice literature.

The paper is structured as follows. Next, Sect. 2 motivates the usefulness of the dominant set selection problem and provides an informal definition. Then, in Sect. 3 we introduce some necessary background on order theory to subsequently formalise the dominant set selection problem in Sect. 4, where we also introduce a simple running example to illustrate the technicalities along the paper. Section 5 outlines the resolution of the DSSP. We base the solution of the DSSP on two operators: lex-cel (in Sect. 6) and anti-lex-cel (in Sect. 7). In Sect. 8 we detail their use to solve the DSSP along with a BIP encoding to

---

[1] Norms provide the means to regulate the behaviour of individuals within a society, and a norm system is a set of norms to enact in that society for regulatory purposes.

[2] Moral values can be described as principles that a society deems valuable.

compute its solution with the aid of state-of-the-art solvers. In more practical terms, Sect. 9 we exploit the tools developed to solve DSSPs to show how to undertake value-aligned norm selection. Section 9 also illustrates how value-aligned norm selection depends on the actual preferences over the value system at hand. Finally, Sect. 10 draws conclusions and sets paths to future work. For the ease of readability, we include a list of notation and symbols after the conclusions.

2 Problem motivation: value-aligned norm selection

As mentioned above, there is a number of problems that require to select the most preferred set of objects considering preferences over their (qualitative) features. Thus, we have discussed that a decision maker may need to choose: students to award grants to; players to form teams; personnel to undertake tasks; projects to be funded; or norms to be enacted.

In fact, that last example will help us to illustrate the characterisation of the problem at hand. Specifically, we assume that there is a set of candidate norms N and we aim to find the set of norms that better aligns with the moral values of the society. Our previous paper [20] introduces some norm examples in an airport border context, where a norm "Permission to cross the border" is aligned with the moral value of "freedom of movement" whereas the norm "Obligation to show passport" is aligned with the value of "security" and is incompatible with the previous norm (i.e., they cannot be simultaneously enacted). Overall, to assess value-alignment we count on a set of moral values, preferences among these values, and a function relating norms to the values that they promote (i.e., specifying norms' features). For instance, consider: four norms $\{n_1, \ldots, n_4\}$; three values $\{v_1, v_2, v_3\}$, being v_1 more preferred than v_2 and v_3, which are indifferent between them; and a feature function that specifies that norm n_1 promotes the three values, and that the remaining norms only promote one value each (n_2 promotes v_1, n_3 promotes v_2 and n_4 promotes v_3). Then, the principle we adhere to is: The more preferred the values promoted by a norm, the more preferred the norm and the more preferred the norms in a set the more value-aligned the set. Thus, we consider $\{n_2\}$ aligns more with moral values than $\{n_4\}$ because n_2 is preferred over n_4 since it promotes a more preferred value. Furthermore, when considering larger sets of these norms, value alignment only grows larger. Following our example, set $S_1 = \{n_1, n_2\}$ is more value-aligned than $S_2 = \{n_3, n_4\}$ because n_1 alone is more preferred than any of the norms in S_2, and adding n_2 only strengthens the value alignment of S_1. Additionally, while the more preferred values have greater impact on assessing which set is more aligned, whenever possible, we still will prefer to select additional norms even if they promote less preferred values (e.g., we favour $\{n_1, n_2, n_3\}$ over S_1). Finally though, since not all norm sets are feasible (norms may be incompatible or redundant between them), the decision maker counts on a function to check if a norm set is feasible or not.

In these terms, the value-aligned norm selection problem consists on finding a set of norms $S \subseteq N$, such that:

- S is feasible;
- S contains the most preferred norms possible (the norms that promote the most preferred values): If we change any norm of S for a more preferred one, the set becomes unfeasible.
- S is *maximal*, namely it is the largest feasible set: adding any further norms to S makes it unfeasible.

We say this S dominates all other feasible sets and therefore we call the problem of finding it *dominant set selection problem* (DSSP). As discussed before many selection problems that count on preferences among the features of the elements can be cast into a DSSP. For example, selecting players to play in a match (where we prefer some types of players to others), awarding research teams (where we prefer to award excellency teams over regular teams), personnel selection (where some capabilities may be preferred over others), etc.

In Sect. 4 below we provide a general formalisation of the dominant set selection problem that encompasses the particular case described above. Before that, we introduce some necessary background on order theory in the following section.

3 Background

Let X be a set of objects. A binary relation $\geq$ on X is said to be: *reflexive*, if for each $x \in X$, $x \geq x$; *transitive*, if for each $x, y, z \in X$, $(x \geq y$ and $y \geq z) \Rightarrow x \geq z$; *total*, if for each $x, y \in X$, $x \geq y$ or $y \geq x$; *antisymmetric*, if for each $x, y \in X$, $x \geq y$ and $y \geq x \Rightarrow x = y$. We can define preferences among the elements of X by means of binary relations. Moreover, we can categorise the type of preferences depending on the properties they hold as follows.

Definition 1 (Preorder, ranking, linear order and partial order) A *preorder* (or *quasi-ordering*) is a binary relation $\geq$ that is reflexive and transitive. A preorder that is also total is said *total preorder* or *ranking*. A total preorder that is also antisymmetric is said a *linear order*. A preorder that is antisymmetric but not total is said a *partial order*.

Note that neither preorders nor rankings are necessarily antisymmetric relations. Thus, given a ranking (or a preorder) $\geq$ if $x, y \in X$, such that $x \geq y$ and $y \geq x$, then we cannot conclude that $x = y$, instead we say these two elements are indifferently preferred and note it as $x \sim y$.

Example 1 Given a set $X = \{x_1, x_2, x_3\}$, an example of ranking would be: $x_1 \geq x_2 \sim x_3$ (with a ranking we know how all elements are related).

Notation 1 We note all possible rankings over X as $\mathcal{R}(X)$.

Using the indifference relation we can consider the quotient set $X/\sim$, which contains the equivalence classes of $\geq$. Thus, given the ranking $x_1 \sim \cdots \sim x_s \geq \cdots \geq x_{r-k} \sim \cdots \sim x_r$, with $x_1, \ldots, x_s, \ldots, x_{r-k}, \ldots, x_r \in X$, then we can consider the quotient set $X/\sim$ with quotient order $\succ$: $\Sigma_1 \succ \cdots \succ \Sigma_n$, where $\Sigma_1 = \{x_1, \ldots, x_s\}, \ldots, \Sigma_n = \{x_{r-k}, \ldots, x_r\} \in X/\sim$ are equivalence classes.

4 Formalising the dominant set selection problem

The goal of this section is to formalise the dominant set selection problem. Informally, and in short, this problem is that of finding a set $S \in \mathcal{P}(X)$ that is both *feasible* and *more preferred than any other set*, and hence *dominates* other sets. Notice that feasibility is an

internal property of each set that captures the compatibility of its elements. However, dominance refers to a preference relation of each set with others that is not initially known since the preferences at hand are those over the features of the elements of a set S. In what follows, we start formally characterising the objects in a dominant set selection problem. Thereafter, we show how to gradually build our formal notion of dominance over sets from the preferences at hand, namely those over features of elements. Finally, we offer a formal definition of the dominant set selection problem.

To start with, we go back to our value-alignment problem in Sect. 2, from which we can generalise to identify the objects that formally characterise the input of a dominant set selection problem as follows:

- a set of elements X;
- a set of features F;
- a ranking $\succeq_F$ over the features in F;
- a function $\mathfrak{f} : X \rightarrow \mathcal{P}(F)$ that outputs the features of each element in X; and
- a feasibility function $\phi : \mathcal{P}(X) \rightarrow \{\top, \bot\}$, which checks if a set $S \in \mathcal{P}(X)$ is feasible ($\phi(S) = \top$ means that it is feasible, and $\phi(S) = \bot$ means that it is not).

We remind the reader that we provide a list of notation and symbols after Sect. 10. At this point, it is important to remark that throughout this paper we consider that $\emptyset$ is not a set in $\mathcal{P}(X)$ ($\emptyset \notin \mathcal{P}(X)$). Therefore, we note as $\mathcal{P}(X)$ the set containing the $2^{|X|} - 1$ different non-empty subsets of X.

From these, informally, solving the dominant set selection problem amounts to selecting a feasible set $S \in \mathcal{P}(X)$ that is more preferred than any other set and includes as many elements as possible. We will say that such set *dominates* the other sets. To select such dominant set we must first formalise our notion of *dominance*. First, we will only consider a single element and define element dominance in: (i) a (equivalence) class of features; and (ii) a whole ranking over features. Once we have established how element dominance works, we will build upon it to define set dominance.

Given a ranking over features $\succeq_F$, we define element dominance within the scope of an equivalence class of features as follows:

Definition 2 Given two elements $x, y \in X$ with features in F, a ranking over features $\succeq_F$, and a feature equivalence class $\Psi \in F/\sim_F$, we say that x is Ψ-dominant over y if

$$|\mathfrak{f}(x) \cap \Psi| > |\mathfrak{f}(y) \cap \Psi|.$$

If $|\mathfrak{f}(x) \cap \Psi| = |\mathfrak{f}(y) \cap \Psi|$, we say that x and y are Ψ-indifferent.

Back to our example in Sect. 2, the dominant set selection problem would be characterised by: $X = \{n_1 \ldots n_4\}$; $F = \{v_1, v_2, v_3\}$; $\mathfrak{f}(n_1) = \{v_1, v_2, v_3\}$, $\mathfrak{f}(n_2) = \{v_1\}$, $\mathfrak{f}(n_3) = \{v_2\}$, $\mathfrak{f}(n_4) = \{v_3\}$; and $v_1 \geq v_2 \sim v_3$. In the quotient order of $F/\sim_F$, this results in two feature equivalence classes: $\Psi_1 = \{v_1\} >_F \Psi_2 = \{v_2, v_3\}$. With this in mind, n_1 is Ψ_1-dominant over n_4 because n_1 promotes v_1 but n_4 does not. n_4 is Ψ_2-dominant over n_2 since n_4 promotes v_3 and n_2 does not promote any value in Ψ_2. Finally n_1 and n_2 are Ψ_1-indifferent as they both promote v_1.

Next, we exploit the definition of element Ψ-dominance to define element dominance considering all the features in F and their ranking $\succeq_F$. Formally:

Definition 3 Given two elements $x, y \in X$ with features in F and a ranking over features $\succeq_F$, we say that x is dominant over y if there is a feature equivalence class $\Psi \in F/\sim_F$, such that:

- x is Ψ-dominant over y; and
- $\forall \Psi' \in F/\sim_F$, such that $\Psi' \succ_F \Psi$, x and y are Ψ'-indifferent.

If neither x dominates y nor vice versa, we say that x and y are indifferent.

Note that the first condition in Definition 3 implies that the dominant element x has more of the features of Ψ than y ($|f(x) \cap \Psi| > |f(y) \cap \Psi|$). As for the second condition, it demands that x and y are indifferent for any other equivalence classes that are more preferred than Ψ. Hence, the most preferred feature equivalence class for which x and y differ, is the class that marks dominance between them.

Back to our example: n_1 is dominant over n_4 because it is Ψ_1-dominant and Ψ_1 is the most preferred feature class; n_1 is also dominant over n_2 because even though they are Ψ_1-indifferent, n_1 is Ψ_2-dominant over n_2.

With the definition of element dominance we now consider dominance between sets in $\mathcal{P}(X)$. Given a set $S = \{x_1, \ldots, x_t\}$, $S \in \mathcal{P}(X)$, we can order its elements in a sequence $(x_{\sigma(1)}, \ldots, x_{\sigma(t)})$ according to dominance, where σ is a permutation of the indexes, such that $\sigma(i)$ is the index in S of the i-th element in the sequence. According to such ordering, $x_{\sigma(i)}$ is indifferent or dominated by $x_{\sigma(1)}, \ldots, x_{\sigma(i-1)}$ while being indifferent or dominating $x_{\sigma(i+1)}, \ldots, x_{\sigma(t)}$. With this in mind we define set dominance as follows.

Definition 4 Given two sets $S = \{x_1, \ldots, x_t\}$ and $S' = \{x'_1, \ldots, x'_r\}$ in $\mathcal{P}(X)$ and their orderings according to dominance $(x_{\sigma(1)}, \ldots, x_{\sigma(t)})$ and $(x'_{\sigma(1)}, \ldots, x'_{\sigma(r)})$ respectively, we say that S is dominant over S' if $\exists j \in \{1, \max(t, r)\}$, such that:

- $x_{\sigma(j)}$ dominates $x'_{\sigma(j)}$ or $j > r$; and
- $x_{\sigma(i)}$ and $x'_{\sigma(i)}$ are indifferent $\forall i < j$.

Notice that the notion of dominance that we propose rewards element *excellence* in a set: the more preferred (excellent) the features of the elements in a set, the more dominant the set. Therefore, a set containing a few excellent elements (with regards to their features) will be preferred over larger sets with *mediocre* elements (i.e. related to less preferred features). This will be the case even if the mediocre elements in a larger set are related to many more features.

Continuing with our example, the set $S_1 = \{n_1, n_2\}$ is *dominant* over $S_2 = \{n_3, n_4\}$ because n_1 is the most dominant element in S_1, n_3 is the most dominant element in S_2, and n_1 is dominant over n_3.

With the definition of set dominance we can now tackle the formalisation of the dominant set selection problem. Formally:

Problem 1 (Dominant set selection problem) Given a set of elements X, a set of features F, a ranking $\succeq_F$ over F, a function $f : X \to \mathcal{P}(F)$ linking the elements in X with their features, and a feasibility function $\phi : \mathcal{P}(X) \to \{\top, \bot\}$ that checks if a set $S \in \mathcal{P}(X)$ is feasible, then the dominant set selection problem (DSSP) is that of finding a set $S \in \mathcal{P}(X)$ such that:

- S is feasible, that is, $\phi(S) = \top$; and

- no other feasible set dominates S, that is, if $S' \in \mathcal{P}(X)$, such that, S' is dominant over $S \Rightarrow \phi(S') = \bot$.

At this point, we remind the reader that our notion of dominance above is meant to reward element excellence. Hence, dominant set selection problems model decision problems for which element excellence is the main decision criterion. This is the case in the examples mentioned in the introduction (granting admissions, committee selection), other examples are awarding prizes or scholarships. Of course, other decision criteria are possible. For example, a decision maker could consider avoiding incompetence as the main decision criterion (in this case elements with more features would be preferred over elements with few more preferred features).

Indeed, it is worth stressing that the problem of how to aggregate different attributes or variables is typically studied in the field of *Multiple Criteria Decision Making* (MCDM), where it is relevant to ask to a decision maker the question on whether the compensation of bad performances on some criteria by good performances on other criteria is acceptable or not [18]. As pointed out in [8], the notion of compensation in general boils down to that of 'tradeoffs' among criteria. For instance, a possibility of compensation is provided by additive utility-based approaches, but there are plenty of other methods offering different levels of compensation, or using non-compensatory aggregation techniques (see, for instance, the article [18] for a discussion about the question guiding to the choice of an appropriate MCDM method and the articles [8, 23] for an axiomatic analysis of MCDM methods in situations with multicriteria non-compensatory preferences; see also [10] for an updated review of compensatory and non-compensatory approaches). Therefore, although this issue is the subject of considerable debate in the MCDM literature, here we define a specific dominance notion that rewards element excellence, and argue that its applicability is strongly dependent on the context.

Note also that the dominant set selection problem may have multiple solutions when multiple sets satisfy the conditions of the problem and do not dominate one another. However, it may also be worth mentioning that, by construction, these solutions will always have the same number of elements (see Sect. 5).

To illustrate the problem and its resolution we use the following problem as a running example in the following sections.

Example 2 Consider four elements $X = \{x_1, x_2, x_3, x_4\}$, two features $F = \{f_1, f_2\}$, the feature ranking $f_1 \succeq_F f_2$ and the feature function $\mathfrak{f}(x_1) = \mathfrak{f}(x_2) = \{f_1\}$ and $\mathfrak{f}(x_3) = \mathfrak{f}(x_4) = \{f_2\}$. In terms of feasibility, we know that any set containing both x_1 and x_3, or both x_2 and x_4, is not feasible (e.g. $\phi(\{x_1, x_2, x_3\}) = \bot, \phi(\{x_3, x_4\}) = \top$). These elements conform an example of dominant set selection problem.

The next section outlines how we actually proceed to solve the dominant set selection problem.

5 Solving the dominant set selection problem: an outline

As we anticipated in the introduction above, we tackle the dominant set selection problem by splitting its resolution in three steps: (1) we extract preferences over single objects based on their features and on the preferences over the features; (2) we rank all possible

sets of objects; and (3) we select the most preferred feasible set of objects. Figure 1 shows the general outline of the steps that we shall follow to solve the dominant set selection problem: *preference grounding*, *preference lifting*, and *feasibility check*. First, preference grounding is performed by grounding the preferences over objects' features to obtain a ranking over the objects in X. Second, preference lifting lifts this element ranking over the elements in X to a set ranking over $\mathcal{P}(X)$. Notice that preference lifting must ensure that the output ranking embodies dominance, thus meaning that a set dominates all its (strictly) less preferred sets in the ranking. Third, the feasibility check step finds the feasible set that is most preferred in the ranking over $\mathcal{P}(X)$. That set will be the set that is dominant over all other feasible sets and, thus, it will constitute the solution to our problem.

The main difficulty when solving the dominant set selection problem lies on generating a ranking over all sets in $\mathcal{P}(X)$. Therefore, the next Sects. 6, 7 and 8 focus on that goal. Sections 6 and 7 introduce two key functions that will allow: (i) to transform a ranking over elements' features into a ranking over elements in X; and (ii) in turn this ranking over elements into a ranking over sets in $\mathcal{P}(X)$.

At this point, we warn the reader that Sect. 6 must be taken as background, since lex-cel was already introduced in [6], whereas Sects. 7 and 8 contain novel contributions.

6 The lex-cel ranking grounding function

The social ranking problem [15] consists on transforming a ranking over $\mathcal{P}(X)$ into a ranking over the elements of X. Thus, a social ranking solution can be viewed as a function $srs : \mathcal{R}(\mathcal{P}(X)) \to \mathcal{R}(X)$, such that for a ranking $\succeq \in \mathcal{R}(\mathcal{P}(X))$, $srs(\succeq) = \succeq_e$ is a ranking of X. Informally, we say that a social ranking solution *grounds* the preferences over subsets to preferences over elements.

Several social ranking solutions have been proposed, such as: a grounding function based on the *ceteris paribus* majority principle [11]; a grounding function based on the notion of marginal contribution [12]; two rankings based on the analysis of majority graphs and minmax score [1]; or the *lex-cel* ranking function [6], which is based on lexicographical preferences. Here, we adapt lex-cel to rank the elements in X based on their features in F.

In more detail, the transformation performed by lex-cel proceeds as follows. First, consider the quotient set $\mathcal{P}(X)/\sim$ (see Sect. 3) such that subsets related by indifference relations fall on the same equivalence class $\Sigma_i \in \mathcal{P}(X)/\sim$. Since the equivalence classes are not indifferent between them, we have a strict quotient order $\succ$ between them: $\Sigma_1 \succ \cdots \succ \Sigma_{|\mathcal{P}(X)/\sim|}$.

We now define a function $\mu : X \to \mathbb{N}^{|\mathcal{P}(X)/\sim|}$, which for an element $x \in X$ returns its profile vector, a natural vector whose dimension is the number of equivalence classes in the quotient set $|\mathcal{P}(X)/\sim|$. The i-th component of the profile vector for x stands for the number of times that x appears in the subsets of equivalence class Σ_i. Notice that equivalence class Σ_i is the class containing the i-th most preferred subsets of $\mathcal{P}(X)$ according to the preorder $\succeq$. For instance, if $\mu(x) = (c_1^x, \dots, c_{|\mathcal{P}(X)/\sim|}^x)$, then c_i^x is the number of times that x appears in the subsets of equivalence class Σ_i. Formally, we define the profile vector for an element $x \in X$ as:

$$\mu(x) = (c_1^x, \dots, c_{|\mathcal{P}(X)/\sim|}^x), \text{ where } c_i^x = |\{S \in \Sigma_i : x \in S\}| \tag{1}$$
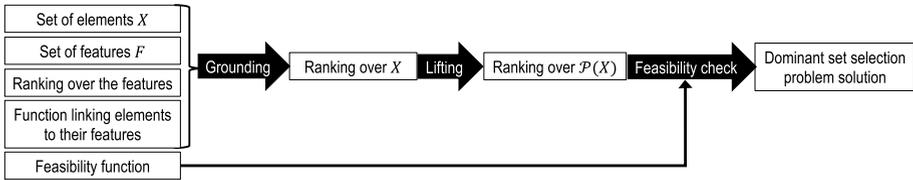
Fig. 1 Outline of the steps to solve the dominant set selection problem

Given any two elements $x, y \in X$, we can establish a preference between them by comparing their profile vectors with the lexicographical order of vectors. That is:

Definition 5 We define the lexicographical order of vectors $\geq_L$ such that given two vectors $c = (c_1, \ldots, c_m), c' = (c'_1, \ldots, c'_m) \in \mathbb{N}^m$, we say that $c >_L c'$ iff $\exists i$, such that $c_1 = c'_1; \ldots; c_{i-1} = c'_{i-1}$ and $c_i > c'_i$. On the other hand, $c =_L c' \Leftrightarrow c = c'$.

We then define the lexicographical-exellence grounded ranking $le(\geq) =\geq_e$ between two elements by comparing their profile vectors. Given $x, y \in X$, we say that:

$$
\begin{cases}
x \succeq_e y \Leftrightarrow \mu(x) \geq_L \mu(y) \\
x \preceq_e y \Leftrightarrow \mu(x) \leq_L \mu(y) \\
x \sim_e y \Leftrightarrow \mu(x) = \mu(y)
\end{cases}
\tag{2}
$$

In [6], the authors prove that grounding preferences with lex-cel satisfies properties that make the grounding fair. In particular, such properties are neutrality, coalitional anonymity, monotonicity and independence of the worst set. Next, we provide a short illustration of these four properties.

First, neutrality ensures that the ranking resulting from applying lex-cel does not depend on the elements' names/identities. Specifically, this property means that if we permute two elements x and y in a ranking $\geq$ over $\mathcal{P}(X)$, the grounded ranking should obey to the same permutation. So, for instance, consider a ranking $\geq$ over $\mathcal{P}(X)$, with $X = \{x, y, z\}$ and such that $\{x, y, z\} \geq \{x\} \geq \{y, z\} \geq \{x, y\} \geq \{y\} \geq \{x, z\} \geq \{z\}$. Suppose that the grounded ranking specifies the relation $x \succeq_e y$ on the ranking $\geq$. Then, the grounded ranking should specify the relation $y \succeq'_e x$ on the ranking $\geq'$ such that $\{x, y, z\} \geq' \{y\} \geq' \{x, z\} \geq' \{x, y\} \geq' \{x\} \geq' \{y, z\} \geq' \{z\}$, which is obtained from $\geq$ by permuting x and y.

Similar to the neutrality property, the coalitional anonymity property extends the anonymity principle to "non-informative" subsets of X: the relative ranking between two elements should only depend on the sequence in which they separately occur along the ranking over $\mathcal{P}(X)$. For instance, in the two rankings $\{x, y, z\} \geq \{x\} \geq \{y, z\} \geq \{x, y\} \geq \{y\} \geq \{x, z\} \geq \{z\}$ and $\{x, z\} \geq' \{y, z\} \geq' \{x, y, z\} \geq' \{x, y\} \geq' \{y\} \geq' \{z\} \geq' \{x\}$, if we focus on sets containing either x or y (but not both), from left to right: first, we have that element x occurs in the singleton set $\{x\}$ in $\geq$ and in the set $\{x, z\}$ in $\geq'$, then element y occurs in the set $\{y, z\}$ in both rankings $\geq$ and $\geq'$, y occurs in the set $\{y\}$ in both rankings $\geq$ and $\geq'$, and finally, x occurs in the subset $\{x, z\}$ in $\geq$ and in $\{x\}$ in $\geq'$. Therefore, since x and y occur according to the sequence x, y, y, x on both rankings $\geq$ and $\geq'$, a grounded ranking satisfying

coalitional anonymity should specify the same relation between x and y on $\succeq$ and $\succeq'$ (i.e., $x \succeq_e y \Leftrightarrow x \succeq'_e y$).

As shown in [6], neutrality and coalitional anonymity together imply that if two elements x and y are such that $\mu(x) = \mu(y)$, then they should be ranked indifferent in the grounded ranking.

A grounded ranking that satisfies monotonicity, breaks possible indifference relations in a consistent way. This means that if on a ranking $\succeq$ over $\mathcal{P}(X)$ a grounded ranking states that two elements x and y are indifferent (i.e, $x \sim_e y$), then, if we consider a new ranking $\succeq'$ obtained from $\succeq$ by improving the position of some subsets containing x but not y, we should have that the grounded ranking ranks x strictly better than y on $\succeq'$ (i.e. $x \succeq'_e y$ and $x \nsim'_e y$). For instance, suppose that on a ranking $\{x, y, z\} \succeq \{x\} \sim \{y, z\} \succeq \{x, y\} \succeq \{y\} \sim \{x, z\} \succeq \{z\}$ the grounded ranking is such that $x \sim_e y$. Now, if we improve the position of the subset $\{x, z\}$, so that we obtain the new ranking $\{x, y, z\} \succeq' \{x\} \sim' \{y, z\} \succeq' \{x, y\} \succeq' \{x, z\} \succeq' \{y\} \succeq' \{z\}$, according to monotonicity we have a grounded ranking such that $x \succeq'_e y$ and $x \nsim'_e y$.

Finally, the property of independence of the worst subsets is aimed at accounting higher ranked subsets over lower ranked ones. Thus, we say that a grounded ranking is independent of the worst subsets if, once the grounded ranking has stated that an element x is strictly better than y, any change in the relative ranking of subsets in the worst indifference class of the ranking over $\mathcal{P}(X)$ does not affect such an assertion. For instance, suppose that on the ranking $\{x, y, z\} \succeq \{x\} \succeq \{y, z\} \succeq \{x, y\} \succeq \{y\} \sim \{x, z\} \sim \{z\}$ the grounded ranking says the $x \succeq_e y$ and $x \nsim_e y$, then it should say the same on $\{x, y, z\} \succeq' \{x\} \succeq' \{y, z\} \succeq' \{x, y\} \succeq' \{y\} \succeq' \{x, z\} \succeq' \{z\}$, which is obtained from $\succeq$ by just modifying the relation among elements of its last equivalence class $\{\{y\}, \{x, z\}, \{z\}\}$. So, giving more importance to occurrences in higher ranked subsets, this property actually rewards the 'excellence' of elements in a ranking over $\mathcal{P}(X)$.

In [6], the authors not only prove that lex-cel satisfies these (logically independent) axioms, but also that it is the only grounding function that satisfies them.

Even though lex-cel is formally defined in [6] as a function $le : \mathcal{R}(\mathcal{P}(X)) \to \mathcal{R}(X)$, here we adapt it to handle the input of the dominant set selection problem and thus perform the grounding process in Fig. 1. Therefore, we redefine lex-cel as a function $le : \mathcal{R}(F) \to \mathcal{R}(X)$. Then, given a ranking of features $f_1 \succeq_F \cdots \succeq_F f_{|F|}$, with quotient order $\Psi_1 >_F \cdots >_F \Psi_{|F/\sim_F|}$ over $F/\sim_F$, and an element $x \in X$, the function μ, would be defined as:

$$\mu(x) = (|\mathsf{f}(x) \cap \Psi_1|, \ldots, |\mathsf{f}(x) \cap \Psi_{|F/\sim_F|}|).$$

Example 3 Following Example 2, note that we know that elements x_1 and x_2 have both the most preferred feature f_1, while x_3 and x_4 have the least preferred feature f_2. With this in mind, their μ vectors would be: $\mu(x_1) = (1, 0)$, $\mu(x_2) = (1, 0)$, $\mu(x_3) = (0, 1)$, $\mu(x_4) = (0, 1)$. Therefore, the grounded ranking over X would be $x_1 \sim_e x_2 \succeq_e x_3 \sim_e x_4$.

7 The anti-lex-cel ranking lifting function

Thanks to lex-cel we can ground a ranking over features in F to a ranking over the elements in X. As shown in Fig. 1, the next step is to lift this ranking over single elements to a ranking over sets of elements, namely over $\mathcal{P}(X)$. This procedure is similar to that of the *ranking*

sets of objects problem surveyed in [3]. The ranking sets of objects problem consists on building a ranking over sets from an ordering over individual elements. Some solutions for the ranking sets of objects problem are maxmin and minmax, as introduced in [2]. Maxmin assesses preferences over sets by comparing only their most preferred element except when these elements are the same, in which case it compares their least preferred elements. On the other hand, minmax does the inverse comparison. It assesses preferences over sets based only on how their least preferred elements are compared. If these elements are the same, the sets' most preferred elements are compared. Note that neither of these methods consider further elements than the most and least preferred ones. This makes them unsuitable for our purpose, since we want to take into account as many elements as possible. The leximin and leximax functions introduced in [16] represent alternative approaches. In summary, leximin and leximax are based on comparing lexicographically sets. In the case of leximin, preferences over sets depend on how their worst elements compare. If these elements are the same, their second worst elements are compared, and so on. If there is no difference, the larger set is preferred (the sets are indifferent if both have the same size). Conversely, leximax compares sets depending on how their best elements compare. If these elements are the same, their second best elements are compared, and so on. If there is no difference, the smaller set is preferred (the sets are indifferent if both have the same size). Unfortunately, we cannot use any of the solutions of the ranking sets of objects problem because they assume a total order of elements. Instead, we have a more general assumption, since we suppose a ranking on elements. Note that this is a crucial difference, since rankings allow for different elements to be indifferently preferred, whereas total orders are antisymmetric, meaning that an element cannot be equally preferred to another element.

Since, to the best of our knowledge, no lifting functions assuming element rankings exist, in this section we formalise a novel one, which we call anti-lex-cel. In Sect. 7.1 we formally introduce anti-lex-cel. Thereafter, in Sect. 7.2 we provide an axiomatic characterisation of anti-lex-cel and we prove that it is the only lifting function satisfying such axioms. Finally, Sect. 7.3 draws the relationship between lex-cel and anti-lex-cel while Sect. 7.4 connects the results in this section with existing results in the literature.

7.1 Formal definition

Anti-lex-cel can be viewed as a function $ale : \mathcal{R}(X) \rightarrow \mathcal{R}(\mathcal{P}(X))$, such that for a ranking $\succeq_e \in \mathcal{R}(X)$, $ale(\succeq_e) = \succeq$ is a ranking over $\mathcal{P}(X)$. We formalise anti-lex-cel in a very similar way to lex-cel, but reversing the process.

To perform anti-lex-cel we start with a ranking $\succeq_e$ over the elements in X. First, we consider the quotient set $X/\sim_e$. Each equivalence class in $X/\sim_e$ contains a set of indifferently preferred elements. Equivalence classes in $X/\sim_e$ are ordered by the quotient order $\succ_e$. Hence, $\Xi_1 \succ_e \cdots \succ_e \Xi_r$, where $r = |X/\sim_e|$ and Ξ_i is the equivalence class containing the i-th most preferred elements. We define a function $\eta : \mathcal{P}(X) \rightarrow \mathbb{N}^r$ to count the appearances of the elements of a set in $\mathcal{P}(X)$ in each equivalence class. Thus, given a set $S \in \mathcal{P}(X)$, $\eta(S)$ is a vector of size r whose i-th component stands for the number of elements in S that are found in the equivalence class Ξ_i. Formally:

$$\eta(S) = (s_1, \ldots, s_r), \text{ where } s_i = |S \cap \Xi_i| \tag{3}$$

Note that, similarly to μ in Eq. 1, $\eta(S)$ is a vector whose elements represent how preferred the elements in S are: the larger the first numbers of the vector, the more preferred the elements in S are (in terms of $\succeq_e$), and hence we can infer that the more preferred S is. This

again means that ranking sets of elements is equivalent to lexicographically ordering their associated vectors as calculated by the η function. Thus, to compare two sets $S, S' \in \mathcal{P}(X)$, we compare lexicographically $\eta(S)$ and $\eta(S')$ (see Definition 5). With those considerations, we are now ready to tackle the formulation of the anti-lex-cel function *ale*. We define $\succeq$ as the ranking of sets in $\mathcal{P}(X)$ such that given two sets $S, S' \in \mathcal{P}(X)$, it orders them according to the following rules:

$$\begin{cases} S \succeq S' \Leftrightarrow \eta(S) \geq_L \eta(S') \\ S \preceq S' \Leftrightarrow \eta(S) \leq_L \eta(S') \\ S \sim S' \Leftrightarrow \eta(S) = \eta(S') \end{cases} \tag{4}$$

After that, we are ready to formally define the anti-lexicographic-excellence ranking lifting function as follows:

Definition 6 Given a set of elements X and a ranking $\succeq_e$ over the elements in X, the ranking lifting function $ale : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ such that $ale(\succeq_e) = \succeq$ is called anti lexicographic excellence (anti-lex-cel).

Example 4 Consider the element ranking $x_1 \sim_e x_2 \succeq_e x_3 \sim_e x_4$ over X that we found in Example 3. We apply anti-lex-cel to this ranking by computing the η vector for the sets in $\mathcal{P}(X)$. Since the quotient order is $\Xi_1 > \Xi_2$, with $\Xi_1 = \{x_1, x_2\}$ and $\Xi_2 = \{x_3, x_4\}$, we have that, for instance, $\eta(\{x_1, x_2, x_3\}) = (2, 1)$ and $\eta(\{x_3, x_4\}) = (0, 2)$. Then, by comparing the η vectors of all sets we can build the following ranking over $\mathcal{P}(X)$: $\{x_1, x_2, x_3, x_4\} \succeq \{x_1, x_2, x_3\} \sim \{x_1, x_2, x_4\} \succeq \{x_1, x_2\} \succeq \{x_1, x_3, x_4\} \sim \{x_2, x_3, x_4\} \succeq \{x_1, x_3\} \sim \{x_1, x_4\} \sim \{x_2, x_3\} \sim \{x_2, x_4\} \succeq \{x_1\} \sim \{x_2\} \succeq \{x_3, x_4\} \succeq \{x_3\} \sim \{x_4\}$.

7.2 Axiomatic characterisation

We now introduce four properties for a ranking lifting function $f : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ and prove that they together axiomatically characterise *ale* and that *ale* is the unique lifting function that satisfies them.

The first axiom is a coherence property saying that the ranking of singleton sets should be "aligned" with $\succeq_e$, where $\succeq_e$ is a ranking of the elements of X.

Axiom 1 (Simple Dominance) Given an element ranking $\succeq_e \in \mathcal{R}(X)$, a ranking lifting function f satisfies the *simple dominance* property iff

$$x \succeq_e y \text{ and } x \nsim_e y \Rightarrow \{x\} \succeq \{y\} \text{ and } \{x\} \nsim \{y\}$$

for all $x, y \in X$ and with $\succeq = f(\succeq_e)$.

The second axiom is an anonymity property: permuting the names of elements should not affect the ranking provided by a lifting function.

Axiom 2 (Neutrality) Given an element ranking $\succeq_e \in \mathcal{R}(X)$, let π be a bijection on X and let $\succeq_e^\pi \in \mathcal{R}(X)$ be such that by

$$x \succeq_e x' \Leftrightarrow \pi(x) \succeq_e^\pi \pi(x')$$

for all $x, x' \in X$. A lifting function f satisfies the *neutrality* property iff

$$S \succeq S' \Leftrightarrow \pi(S) \succeq^\pi \pi(S')$$

for all $S, S' \in \mathcal{P}(X)$ and where $\pi(S)$ and $\pi(S')$ are the images of S and S' through π and where $\succeq = f(\succeq_e)$ and $\succeq^\pi = f(\succeq_e^\pi)$.

The next axiom says that if a set S is (weakly) preferred to another one S', then adding new elements to the preferred one S makes this new set (strictly) preferred to S'.

Axiom 3 (Size Monotonicity) Given an element ranking $\succeq_e \in \mathcal{R}(X)$, a ranking lifting function f satisfies the *size monotonicity* property iff

$$S \succeq S' \Rightarrow (S \cup \bar{S}) \succ S' \text{ and } (S \cup \bar{S}) \not\sim S'$$

for all $S, S' \in \mathcal{P}(X)$ and $\bar{S} \subseteq (X \setminus S)$, $\bar{S} \neq \emptyset$, with $\succeq = f(\succeq_e)$.

The next axiom aims at rewarding the best elements preventing the overestimation of dominated ones and states that a strict preference between two sets S and S', i.e. $S \succeq S'$ and $S \not\sim S'$, should not be affected by the addition of new single element that are strictly worse (with respect to the element ranking $\succeq_e$ of X) to those already contained in the preferred set S.

Axiom 4 (Independence of the Worst Elements) Given an element ranking $\succeq_e \in \mathcal{R}(X)$, a ranking lifting function f satisfies the *independence of the worst elements* property iff

$$S \succeq S' \text{ and } S \not\sim S' \Rightarrow S \succeq (S' \cup \bar{S}') \text{ and } S \not\sim (S' \cup \bar{S}')$$

for all $S, S' \in \mathcal{P}(X)$ and $\bar{S}' \subseteq (X \setminus S')$, $\bar{S}' \neq \emptyset$, such that $x \succeq_e x'$ and $x \not\sim_e x'$ for all $x \in S$ and $x' \in \bar{S}'$ and with $\succeq = f(\succeq_e)$.

The following proposition establishes that anti-lex-cel satisfies the four axioms above.

Proposition 1 *The anti-lex-cel lifting function ale satisfies Axioms* 1, 2, 3 *and* 4.

Having axiomatized anti-lex-cel, we can obtain a stronger result. Thus, the following theorem tells us that in fact anti-lex-cel is the only lifting function that satisfies these axioms.

Theorem 1 *Let* $f : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ *be a ranking lifting function. Then* f *satisfies Axioms* 1, 2, 3 *and* 4 *if and only if* f *is the anti-lex-cel lifting function ale.*

For the sake of readability, we detail the proofs of Proposition 1 and Theorem 1 in Appendix A.1.

7.3 On the relation between lex-cel and anti-lex-cel

As noticed above, the anti-lex-cel function is very similar to lex-cel, though it real-ises the reverse process (from ranking over elements to ranking over sets of elements). However, notice that, since $le : \mathcal{R}(\mathcal{P}(X)) \rightarrow \mathcal{R}(X)$ is not injective (it cannot be because $|\mathcal{R}(\mathcal{P}(X))| > |\mathcal{R}(X)|$), there is no inverse for lex-cel, and therefore, in general, anti-lex-cel is not the inverse of lex-cel. Nonetheless, in what follows we characterise the conditions under which anti-lex-cel becomes the inverse of a restriction of lex-cel.

Before establishing such formal result, we introduce an auxiliary result that will prove useful for that purpose. The following lemma states that we can build the profile vector for an element in a compositional, additive manner. More precisely, we can obtain the μ and η profile vectors by adding up the profile vectors restricted to each part in a partition of $\mathcal{P}(X)$. This property will help us prove the relation between lex-cel and anti-lex-cel in the forth-coming Theorem 2.

Lemma 1 *Given* $P_1, \dots, P_k$ *a partition of* $\mathcal{P}(X)$:

$$\mu(x) = \mu|_{P_1}(x) + \cdots + \mu|_{P_k}(x), \forall x \in X$$

$$\eta(S) = \eta|_{P_1}(S) + \cdots + \eta|_{P_k}(S), \forall S \in \mathcal{P}(X)$$

where

- $\mu|_{P_j}(x) = (c_1, \dots, c_l)$, with $c_i = |\{S \in \Sigma_i \cap P_j : x \in S\}|$, stands for the profile vector of element x restricted to partition P_j; and
- $\eta|_{P_j}(S) = (s_1, \dots, s_r)$, with $s_i = |(\{S\} \cap P_j) \cap \Xi_i|$, stands for the profile vector of set S restricted to partition P_j.

Proof 1 The proof is straightforward considering that μ and η are vectors of cardinali-ties and cardinalities satisfy that if $S \subseteq X$, and $P_1, \dots P_k \subseteq X$ is a partition of X, then $|S \cap X| = \sum_i |S \cap P_i|$.

The following result tells us that given a ranking $\succeq_e$ over the elements of X, the compo-sition of anti-lex-cel and lex-cel over it results in the very same ranking $\succeq_e$.

Theorem 2 *Given a ranking* $\succeq_e \in \mathcal{R}(X)$, $le(ale(\succeq_e)) = \succeq_e$.

Proof 2 Suppose $ale(\succeq_e) = \succeq$ and $le(ale(\succeq_e)) = \succeq'_e$. First, note that if $\succeq_e$ is such that $\forall x, y \in X, \ x \sim_e y$, then when applying ale to $\succeq_e$ we would have that $\forall S \in \mathcal{P}(X)$, $\eta(S) = (|S|)$ which would mean that the preference of a set only depends on its car-dinality (not on its elements), and when applying back le to the obtained set rank-ing we would have that $\forall x, y \in X, \mu(x) = \mu(y)$ as all elements in X appear in the same number of sets of a certain cardinality. Therefore, to prove the theorem we can con-sider that $x, y \in X$, such that $x \succeq_e y$ and $x \nsim_e y$ and prove that $x \preceq'_e y$ is not pos-sible. Now consider $XYS = \{S \in \mathcal{P}(X), x, y \in S\}$, $XS = \{S \in \mathcal{P}(X), x \in S, y \notin S\}$, $YS = \{S \in \mathcal{P}(X), x \notin S, y \in S\}$ and $RS = \{S \in \mathcal{P}(X), x, y \notin S\}$, note that these subsets form a partition of $\mathcal{P}(X)$ ($\mathcal{P}(X) = XYS \cup XS \cup YS \cup RS$, and XYS, XS, YS, RS disjoint), thus when

applying le $(le(\geq) = \geq_e)$ we will have: $\mu(x) = \mu|_{XYS}(x) + \mu|_{XS}(x) + \mu|_{YS}(x) + \mu|_{RS}(x)$ and $\mu(y) = \mu|_{XYS}(y) + \mu|_{XS}(y) + \mu|_{YS}(y) + \mu|_{RS}(y)$, and $\mu|_{XYS}(x) = \mu|_{XYS}(y), \mu|_{RS}(x) = \mu|_{RS}(y)$, and $\mu|_{YS}(x) = \mu|_{XS}(y) = (0, \dots, 0)$. Since, $y \succeq'_e x$, we have that $\mu(y) \geq_L \mu(x)$ which considering the equalities above implies that $\mu|_{YS}(y) \geq_L \mu|_{XS}(x)$, since $|YS| = |XS|$, this would mean that $\exists S \in YS$, such that $\forall S' \in XS$, $S \geq S'$ or alternatively $\eta(S) \geq_L \eta(S')$. But this is not possible, consider $S' = S \setminus \{y\} \cup \{x\}$, this set contains x, therefore $S' \in XS$ and since $x \succeq_e y$ and $x \nsim_e y, \eta(S') >_L \eta(S)$, which proves the theorem.

Based on the theorem above, we can establish that anti-lex-cel is the inverse of lex-cel for a restricted family of rankings $ILE = \{\geq \in \mathcal{R}(\mathcal{P}(X)), \exists \geq_e \in \mathcal{R}(X) ale(\geq_e) = \geq\}$.

Corollary 1 $le|_{ILE}$ is the inverse of ale.

Proof 3 In Theorem 2 we have seen that $le(ale(\geq_e)) = \geq_e$, which means that $le|_{ILE}(ale(\geq_e)) = \geq_e$ as we are only restricting the domain of le, now due to this restriction le is injective and exhaustive, as is ale, so they are inverses.

7.4 Related results from the literature

Next we investigate the relationship between anti-lex-cel and a related result in the literature. In [7], Bossert et al. study a particular preorder on $\mathcal{P}(X)$ associated to a linear order on X. Therefore, analogously to anti-lex-cel, [7] studies a lifting of preferences from the element level (the linear order) to the set level. Interestingly, in this section we show that when fed with a linear order on X (which is a particular type of ranking), the output of anti-lex-cel is precisely the preorder on $\mathcal{P}(X)$ studied in [7]. Hence, this shows the generality of anti-lex-cel.

Given an element ranking $\geq_e \in \mathcal{R}(X)$ that is also anti-symmetric (i.e., $\geq_e$ is a linear order), in [7] the authors have introduced the following properties for a preorder (a transitive and reflexive relation) $\geq$ of $\mathcal{P}(X)$ associated to $\geq_e$ (see also [3] for a general review of the related literature):

- Simple dominance (SD): for any $x, y \in Y$, $x \succeq_e y$ and $x \nsim_e y \Rightarrow \{x\} \geq \{y\}$ and $\{x\} \nsim \{y\}$;
- Simple Monotonicity (SM): for any $x, y \in X$ with $x \neq y$, $\{x, y\} \geq \{x\}$ and $\{x, y\} \nsim \{x\}$;
- Independence (IND): for any $S, T \in \mathcal{P}(X)$, for each $x \in X \setminus (S \cup T)$, $S \geq T \Leftrightarrow S \cup \{x\} \geq T \cup \{x\}$;
- Robustness for strict Preferences (RP): for any $S, T \in \mathcal{P}(X)$, for each $x \in X \setminus (S \cup T)$

$$\left. \begin{array}{l} S \geq T \text{ and } S \nsim T, \\ y \succeq_e x \text{ and } y \nsim_e x \; \forall y \in S, \\ z \succeq_e x \text{ and } z \nsim_e x \; \forall z \in T \end{array} \right\} \Rightarrow S \geq T \cup \{x\} \text{ and } S \nsim T \cup \{x\}.$$

A particular preorder on $\mathcal{P}(X)$ associated to a linear order $\geq_e$ on X has been studied in [7]. To define it, we need some more notations. Without loss of generality, it is assumed that the elements of any set $S = \{x_1, \dots, x_s\} \in \mathcal{P}(X)$ are ordered in decreasing preference according to $\geq_e$, that is, $x_1 \succeq_e x_2 \succeq_e \dots \succeq_e x_s$.

Let $u_{\succeq_e} : X \to \mathbb{R}_{>0}$ be a real-valued function such that for all $x, y \in X$, $u_{\succeq_e}(x) \geq u_{\succeq_e}(y) \Leftrightarrow x \succeq_e y$. For $S = \{x_1, \ldots, x_s\} \in \mathcal{P}(X)$, let $v(S)$ be an $|X|$-dimensional vector such that

$$v_{u_{\succeq_e}}(S) = (u_{\succeq_e}(x_1), \ldots, u_{\succeq_e}(x_s), 0, \ldots, 0),$$

so, the last $|X| - |S|$ components of the vector are completed with zeros.

The relation $\succeq_{u_{\succeq_e}}$ on $\mathcal{P}(X)$ is then defined as follows:

$$S \succeq_{u_{\succeq_e}} T \Leftrightarrow v_{u_{\succeq_e}}(S) \geq_L v_{u_{\succeq_e}}(T).$$

for all $S, T \in \mathcal{P}(X)$. The following result, which has been proved in [7], states that $\succeq_{u_{\succeq_e}}$ is the unique preorder of $\mathcal{P}(X)$ that satisfies the four properties.

Theorem 3 [7] *Let $\succeq$ be a preorder on $\mathcal{P}(X)$. $\succeq$ satisfies SD, SM, IND and RP iff $\succeq = \succeq_{u_{\succeq_e}}$.*

We now prove that the total preorder $\succeq = ale(\succeq_e)$ also satisfies properties SD, SM, IND and RP.

Proposition 2 *Given a linear order $\succeq_e \in \mathcal{R}(X)$, the ranking $\succeq = ale(\succeq_e)$ satisfies SD, SM, IND and RP.*

Proof 4 From Proposition 1 we know that f satisfies Axioms 1, 3 and 4.

From Axiom 1 on *ale*, we directly have that $\succeq$ satisfies SD. Since $\succeq$ is total, we have $\{x\} \sim \{x\}$.

Then, the proof that $\succeq$ satisfies SM follows by Axiom 3 on *ale* with $\{x\}$ in the role of both S and S' and $\bar{S} = \{y\}$.

To prove IND, simply notice that for all $S, T \in \mathcal{P}(X)$, $\eta(S) \geq_L \eta(T) \Leftrightarrow \eta(S \cup \{x\}) \geq_L \eta(T \cup \{x\})$ for all $x \in X \setminus (S \cup T)$.

Finally, the proof that $\succeq$ satisfies RP follows by Axiom 4 on *ale* with T in the role of S' and $\bar{S'} = \{x\}$.

To end this section, the following corollary formally establishes the relationship between anti-lex-cel and the results by Bossert et al. in [7].

Corollary 2 *Let $\succeq_e$ be a linear order on X, then $\succeq = \succeq_{u_{\succeq_e}}$ where $\succeq = ale(\succeq_e)$.*

Proof 5 The proof follows directly from Theorem 3 and Proposition 2.

8 Solving the dominant set selection problem

With both lex-cel and anti-lex-cel, we can now address solving the dominant set selection problem. As shown in Fig. 1, we will build the solution through three steps. In particular we will transform the input of the DSSP into a ranking over X using lex-cel, then we use anti-lex-cel to obtain a ranking over $\mathcal{P}(X)$. Thanks to the properties of *ale*, this ranking embodies dominance as in Definition 4, meaning that a set is dominant over its least preferred sets in the ranking. With this ranking and the feasibility function, we can find the solution as the more preferred set in the ranking that is feasible.

We start with the elements in X, the set of features F, their ranking $\succeq_F$ and the feature function $\mathfrak{f}$ relating elements to their features. In Sect. 6, we have adapted the lex-cel function to ground the ranking $\succeq_F$ to a ranking over X such that $le(\succeq_F) = \succeq_e$. Then, we apply anti-lex-cel to this ranking, $ale(\succeq_e) = \succeq$, to obtain a ranking over $\mathcal{P}(X)$. Thus, by composing lex-cel and anti-lex-cel, we can define a function $dom : \mathcal{R}(F) \rightarrow \mathcal{R}(\mathcal{P}(X))$ that transforms a ranking over the features in F to a ranking over the sets in $\mathcal{P}(X)$ as $dom(\succeq_F) = ale(le(\succeq_F))$. We show that the resulting ranking from the *dom* function embodies dominance as stated by the following theorem.

Theorem 4 *Let X be a set of elements, F a set of features, $\succeq_F$ a ranking over F, and a function $\mathfrak{f}$ relating elements to their features. For any pair $S, S' \in \mathcal{P}(X)$, S is dominant over S' $\Leftrightarrow S \succ S'$ ($S \succeq S'$ and $S \nsim S'$), where $dom(\succeq_F) = ale(le(\succeq_F)) = \succeq$.*

For the sake of readability, we detail the proof of Theorem 4 in the appendix (see Sect. A.2).

Corollary 3 *Consider a dominant set selection problem with a set of elements X, a set of features F, a ranking $\succeq_F$ over F, and a function $\mathfrak{f}$ relating elements to their features. Consider a set $S_{pref} \in \mathcal{P}(X), \phi(S_{pref}) = \top$, such that $\forall S' \in \mathcal{P}(X)$ with $S' \succeq S_{pref}$ and $S' \nsim S_{pref} \Rightarrow \phi(S') = \bot$. Then, S_{pref} is a solution to the dominant set selection problem.*

Proof 6 This result follows directly from Theorem 4.

With this result, note that to find the solution to the dominant set selection problem, the only step left to do after building $\succeq$ is to check for feasibility from the most preferred set in $\succeq$ to the least preferred set in $\succeq$ until we find the most preferred one that is feasible.

Nonetheless, note that building the ranking $dom(\succeq_F) = \succeq$ to solve the dominant set selection problem turns out to be rather costly. It requires to compute the η profile vector in Eq. 3 for every subset in $\mathcal{P}(X)$, with cost $O(2^{|X|})$, to subsequently order them following Eq. 4, which requires $O(2^{|X|} \cdot log(2^{|X|}))$ in the average case ($O(2^{2|X|})$ in the worst case). Therefore, finding the solution has worse than exponential complexity on the number of elements of X, hence hindering applicability.

With the intent of solving the DSSP through optimisation techniques, we show an alternative way of comparing sets of $\mathcal{P}(X)$ avoiding the cost of explicitly building $\succeq$. In particular, we propose a function, the so-called *preference function* $\mathfrak{p} : \mathcal{P}(X) \rightarrow \mathbb{N}$, which embodies the preferences in the $\succeq$ ranking while not needing to build it. Given a set $S \in \mathcal{P}(X)$, the larger its value by the preference function, the more preferred it is in $\succeq$. Importantly, we prove that this function adheres to $\succeq$, meaning that for all pairs of sets, the ranking between each pair is maintained by the function's output.

Let $S \in \mathcal{P}(X)$ be a set of elements. We define its profile vector as $\eta(S) = (c_1^S, \ldots, c_r^S)$, where $c_i^S = |S \cap \Xi_i|$ and $r = |X/\sim_e|$. From that, we compute the preference value of S as follows:

$$\mathfrak{p}(S) = \sum_{i=1}^{r} |S \cap \Xi_i|(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1), \text{ where } \mathfrak{p}(\Xi_r) = |\Xi_r|. \tag{5}$$

Recall that Ξ_r is the equivalence class containing the least preferred elements of X. Notice that by applying Eq. 5, we can compute the preference of each equivalence class Ξ_i:

$\mathfrak{p}(\Xi_i) = |\Xi_i|(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1)$. Hence, the preference of the classes in the quotient order $\Xi_1 >_e \cdots >_e \Xi_r$ can be recursively computed starting from Ξ_r. Note also that $\mathfrak{p}(S) \geq 0$, and $\mathfrak{p}(S) \in \mathbb{N}$ for any S.

The $\mathfrak{p}$ preference function embodies the ranking $\geq$ over sets in $\mathcal{P}(X)$, as we now prove through the following theorem.

Theorem 5 *Given two sets $S, S' \in \mathcal{P}(X)$, $S \geq S' \Leftrightarrow \mathfrak{p}(S) \geq \mathfrak{p}(S')$.*

Again, for the sake of readability, we detail the proof of Theorem 5 in the appendix (see Sect. A.3).

The preference function $\mathfrak{p}$ together with the results in Theorems 4 and 5 are key to cast the dominant set selection problem as the optimisation problem expressed by the following corollary.

Corollary 4 *Consider a dominant set selection problem with a set of elements X, a set of features F, a ranking $\succeq_F$ over F, and a function $\mathfrak{f}$ relating elements to their features. A feasible set $S_{max} \in \mathcal{P}(X)$ with maximum preference $\mathfrak{p}$ (see Eq. 5):*

$$S_{max} = \arg \max_{S \in \mathcal{P}(X),\, \phi(S) = \top} \mathfrak{p}(S) \tag{6}$$

is a solution to the dominant set selection problem.

Proof 7 This result follows directly from Theorems 4 and 5 and Corollary 3.

Building the whole ranking or computing the preference of all possible subsets is computationally costly. Nonetheless, in those cases in which the feasibility function can be translated into linear or quadratic constraints we can profit from the preference function $\mathfrak{p}$ to encode the DSSP into a binary integer program (BIP) and solve it with state of the art solvers. Thus, hereafter we will assume that we can translate the feasibility function into a set of linear or quadratic constraints C. The first step to encode the dominant set selection problem is to build the objective function of the BIP. The challenge here is to compactly represent the sets of $\mathcal{P}(X)$. Notice that for $X = \{x_1, x_2, x_3\}$, the set $S = \{x_1, x_2\}$ can be represented as $\{x_1, x_2, \neg x_3\}$, or as the binary vector $(1, 1, 0)$. In general, any $S \in \mathcal{P}(X)$ can be encoded as a vector $(d_1, \ldots, d_{|X|})$, where $d_i \in \{0, 1\}$ is the decision variable for element $x_i \in X$: if $d_i = 1$ means that x_i is in S, while $d_i = 0$ means x_i is not in S. Using the $(d_1, \ldots, d_{|X|})$ encoding for sets and following Eq. 5, in general we can obtain the preference of any set as $\sum_{i=1}^{r}(\sum_{x_w \in \Xi_i} d_w)(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1)$, making use of the fact that $|S \cap \Xi_i| = \sum_{x_w \in \Xi_i} d_w$. Therefore, solving Problem 1 amounts to finding the assignment of variables $(d_1, \ldots, d_{|X|})$ representing a feasible set with maximum preference. For that, we propose to solve the following BIP:

$$\max \sum_{i=1}^{r} \left(\sum_{x_w \in \Xi_i} d_w \right) \left(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1 \right) \tag{7}$$

We require that the selected set satisfies the constraints in C. Thus, we consider these constraints in the encoding.

Observe that our BIP employs $|X|$ binary decision variables ($d_i \in \{0, 1\}$) and avoids the expensive, explicit computation of the ranking. Instead, it only requires to compute the preference of the equivalence classes $\Xi_i(\mathfrak{p}(\Xi_i))$. Since our objective function is always

linear, if the constraints in C are either linear or quadratic, we can resort to off-the-shelf integer programming solvers like CPLEX or Gurobi. If the constraints in C are linear we would have to solve a typical BIP, whereas if they are quadratic we would have to solve a Binary Integer Quadratically Constrained Program. Appendix B details the algorithms to build the BIP and provides a link to an implementation.

In the next section we show the BIP that results when encoding the value-alignment norm selection problem introduced in Sect. 2. There we show an example of objective function together with a collection of linear constraints.

9 Application: value-aligned norm selection

Now we count on the tools to solve the dominant set selection problem. Hereafter we will revisit the value-alignment problem described in Sect. 2 to exploit such tools for solving it, hence helping the decision maker.

Recall that in our value-alignment problem the decision maker is presented with a set of candidate norms N and a value system, which includes moral values and their preferences. Each of the norms in N is linked to some values, meaning that each norm promotes the values it is linked to. The problem for the decision maker is to select the subset of norms in N that better aligns with the values.

Prior to casting the problem faced by the decision maker as a particular type of dominant set selection problem, we must formally characterise: (i) the elements (norms); and (2) the features and preferences over features (value system).

First we start with the elements. For that, we base the definition of norms and their relationships in those in [22]. Hereafter we consider that norms are meant to regulate a multi-agent system composed of a set of agents Ag, with a finite set of actions $\mathcal{A}$ available to them.

Furthermore, we consider a simple, first-order language $\mathcal{L}$ to describe the state of the multi-agent system. With these definitions in place, we formalise the elements of our problem, the norms:

Definition 7 (Norm) A norm is a pair $\langle \varphi, \theta(a) \rangle$, where φ is a precondition in the language $\mathcal{L}$, $a \in \mathcal{A}$ is the regulated action, and $\theta \in \{Obl, Per, Prh\}$ is a deontic operator.

Example 5 Say that a country has to decide the norms to apply to its airport borders. The following norms are considered: n_1 permits to cross the border, n_2 prohibits to scan the baggage, n_3 obliges to show a passport, and n_4 obliges to scan the baggage. These norms can be represented formally as follows: n_1 as $\langle \emptyset, \text{Per(cross)} \rangle$; n_2 as $\langle \emptyset, \text{Prh(scan-bag)} \rangle$; n_3 as $\langle \emptyset, \text{Obl(show-passport)} \rangle$; and n_4 as $\langle \emptyset, \text{Obl(scan-bag)} \rangle$. Since we will use these norms in following examples, for the sake of readability we will note them omitting their precondition.

Given a set of norms N, relationships between norms may hold. Thus, we identify norm exclusivity and generalisation as norm relations. Such relationships are relations over norms, henceforth noted as R_x and R_g respectively. Two norms n, n' are *mutually exclusive*, noted as $(n, n') \in R_x$, when they cannot be enacted at once; and they have a *direct generalisation* relation, noted as $(n, n') \in R_g$, when n is more general than n' and there is no other $n_{mid} \in N$, such that n is more general than n_{mid} being n_{mid} more general than n'. We note $A(n)/S(n)$ the ancestors/successors of n.

By putting together norms and their relations, we fully characterise the normative dimension of our decision space.

Definition 8 A norm net is a structure $\langle N, R \rangle$, where N is a set of norms and $R = \{R_x, R_g\}$ is the set of exclusive, and generalisation relations.

Likewise [22], henceforth we shall refer to any subset $\Omega \subseteq N$ as a *norm system*. We are interested in a particular type of norm systems: those that contain neither conflicting nor redundant norms. Thus, we characterise norm systems that avoid both conflicts and redundancy as *sound* norm systems.

Definition 9 Given a norm net $\langle N, R \rangle$, a norm system $\Omega \subseteq N$ is sound iff it is both conflict-free and non-redundant, that is a norm system $\Omega \subseteq N$ is sound if for each $n_i, n_j \in \Omega$, $(n_i, n_j) \notin R_x$; $n_j \notin A(n_i)$; and $\forall n \in N$, such that $|\bar{S}(n)| > 1$, then $\bar{S}(n) \nsubseteq \Omega$, where $\bar{S}(n)$ are the direct successors of n ($\bar{S}(n) = \{n' \in N, (n, n') \in R_g\}$).

At this point, notice that sound norm systems represent *feasible* norm systems. Therefore, when casting our value-alignment problem as a dominant set selection problem, checking for feasibility would consist in checking for soundness.

Example 6 Consider the norms in Example 5, note that we cannot jointly allow to cross the border freely while obliging to show a passport, therefore n_1 and n_3 are incompatible norms. On the other hand, we cannot both oblige to scan a bag and prohibit it, making norms n_2 and n_4 incompatible as well. Thus, the norm net for the norms in Example 5 is the one in Fig. 2

The features in this particular instance of a dominant set selection problem are values. Ethical reasoning typically involves a *value system*, that contains a set of moral values, which are principles that the society deems valuable. As noted in [5], within a value system, some values are preferred to others, and such preferences over moral values influence decision making. Therefore, the preferences over the moral values of a value system, together with the values themselves, have been identified as a core component for ethical reasoning in [5, 14, 21]. Formally,

Definition 10 A value system is a pair $\langle V, \succeq_v \rangle$, where V stands for a non-empty set of values, and $\succeq_v$ is a ranking over the moral values in V.

The definition of value system contains a ranking over moral values, and hence this is the ranking over features.

As required by the dominant set selection problem, we define a function linking the norms (elements) to their values (features). Note though that due to the interplay between values and norm relations, this function must fulfil some conditions. Thus:

Definition 11 Given a norm net $\langle N, R \rangle$ and a value system $\langle V, \succeq_v \rangle$, we call value promotion function the function $\mathfrak{f} : N \rightarrow \mathcal{P}(V)$ that for each norm returns the set of values the norm promotes $\mathfrak{f}(n)$.

Fig. 2 Example of candidate norms for border control along with their relations and their promotion of the free movement and safety values
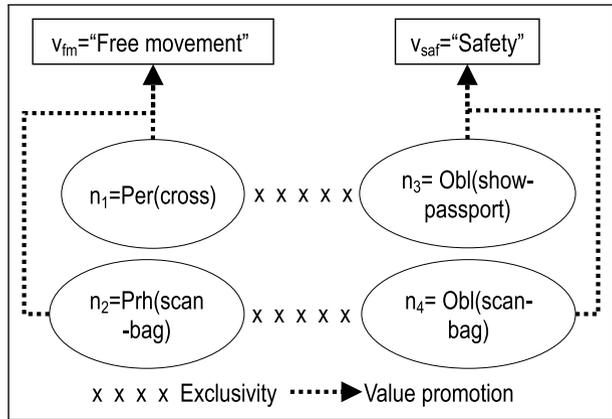
In norm selection, a norm that does not promote any value and a value that is not promoted by any norm are irrelevant. Henceforth, we suppose that all norms promote at least one value ($\forall n \in N, \mathfrak{f}(n) \neq \emptyset$), and that all values are promoted by at least one norm ($\forall v \in V, \exists n \in N$, s.t. $v \in \mathfrak{f}(n)$).

Example 7 Following Example 6, we observe that n_1 and n_2 promote free movement of people/goods ($\mathfrak{f}(n_1) = \mathfrak{f}(n_2) = \{v_{fm}\}$), whereas the rest of norms promote safety ($\mathfrak{f}(n_3) = \mathfrak{f}(n_4) = \{v_{saf}\}$), as depicted in Fig. 2.

With the definitions of the various structures of norms and values we can now define the problem faced by the decision maker that we want to solve.

Problem 2 Given a norm net $\langle N, R \rangle$, a value system $\langle V, \succeq_v \rangle$ and a value promotion function $\mathfrak{f}$, we call value-aligned norm selection (VANS) problem, the problem of finding the set of norms $S \in \mathcal{P}(N)$, such that S is a sound norm system and any other norm system S', that dominates S is not sound.

The value-aligned norm selection problem is a particular instance of the dominant set selection problem.

To solve the value-aligned norm selection problem, we proceed as detailed in Sect. 8. First, we apply lex-cel to the value ranking (as we have done in Example 3).

Once we obtain the ranking over N, we would just apply *ale* to obtain the ranking over all possible norm systems (as done in Example 4).

With the ranking over all norm systems in $\mathcal{P}(N)$, it remains to check for feasibility (in this case by checking for soundness).

Example 8 An example value-aligned norm selection problem would be that where N are the norms in Example 5, with the relations R in Example 6 to assess feasibility and the value promotion function $\mathfrak{f}$ defined in Example 7, supposing the value system $V = \{v_{fm}, v_{saf}\}$, with the value ranking (feature ranking) $v_{fm} \succeq_v v_{saf}$. Note that this structure is completely equivalent to the dominant set selection problem formulated in Example 2, therefore in this case we already know the $\mathcal{P}(N)$ ranking as we have found it in Example

4. Thus, the solution to this value-aligned norm selection problem is $\{n_1, n_2\}$ because it is the first sound (feasible) norm system in the ranking. Note that all norm systems with 3 or 4 norms contain a pair of exclusive norms and out of all the norm systems with 2 norms, $\{n_1, n_2\}$ is the most preferred one. In conclusion, we provided some norms to regulate an airport and by preferring freedom of movement of people/goods over security we selected the norms allowing to cross the border freely to both people and their belongings.

Nonetheless, as explained previously, the exhaustive approach followed above is computationally expensive. Instead, we can solve a VANS problem as an optimisation problem by encoding it into a BIP, as explained in Sect. 8. Building the objective function for this encoding is straightforward from Eq. 7. We must simply consider that there is one decision d_i variable for each norm $n_i \in N$. Moreover, we must add the following constraints to ensure that the resulting solution is feasible (the resulting norm system is sound):

- Mutually exclusive (incompatible) norms cannot be selected at once:

$$d_i + d_j \leq 1 \text{ for each } (n_i, n_j) \in R_x \tag{8}$$

- A norm cannot be simultaneously selected with any of its ancestors:

$$d_i + d_k \leq 1 \text{ for each } n_k \in A(n_i) \ \ 1 \leq i \leq |N| \tag{9}$$

- If a norm has more than one direct successor (we note $\bar{S}(n) = \{n' \in N, (n, n') \in R_g\}$), these direct successors cannot be simultaneously selected:

$$\text{If } |\bar{S}(n)| > 1 \text{ then} \sum_{n_j \in \bar{S}(n)} d_j < |\bar{S}(n)| \quad \text{for each } n \in N \tag{10}$$

Algorithms to build the BIP for a VANS problem and a link to an implementation can be found in Appendix C.

Example 9 Example 8, details a value-aligned norm selection problem and how to build a norm system ranking for it. Next we provide the BIP encoding for this example problem. First, we will build our objective function. Since the element ranking is $n_1 \sim_e n_2 \succeq_e n_3 \sim_e n_4$ (see Example 3), the quotient order is $\Xi_1 >_e \Xi_2$, we first compute $\mathfrak{p}(\Xi_2) = |\Xi_2| = 2$, because $\Xi_2 = \{n_3, n_4\}$ (we can also compute $\mathfrak{p}(\Xi_1) = |\Xi_1| \cdot (\mathfrak{p}(\Xi_2) + 1) = 6$, because $\Xi_1 = \{n_1, n_2\}$, though we do not need this number). Therefore, the objective function (following Eq. 7) which we want to maximise is $3d_1 + 3d_2 + d_3 + d_4$. Since the norms of our running example have some relations between them, as shown in Fig. 2, we consider the following constraints regarding exclusive norms: $d_1 + d_3 \leq 1$, $d_2 + d_4 \leq 1$. With this encoding the solution to the BIP is $\{n_1, n_2\}$ (the same we found in Example 8 using $\succeq$).

Different value rankings may vary the selection of the value-aligned norm system. In previous examples we solved the problem of norm selection in an airport depicted in Fig. 2 by considering the value ranking (feature ranking) $v_{fm} \succeq_v v_{saf}$. Subsequent Examples 10 and 11 explore how the solution changes for alternative value rankings ($v_{fm} \preceq_v v_{saf}$ and $v_{fm} \sim_v v_{saf}$).

Example 10 Supposing $v_{fm} \preceq_v v_{saf}$, when grounding these preferences with lex-cel we obtain the norm ranking $n_3 \sim_e n_4 \succeq_e n_1 \sim_e n_2$. Then, lifting this ranking with anti-lex-cel

we obtain: $\{n_1, n_2, n_3, n_4\} \succeq \{n_1, n_3, n_4\} \sim \{n_2, n_3, n_4\} \succeq \{n_3, n_4\} \succeq \{n_1, n_2, n_3\} \sim \{n_1, n_2, n_4\}$ $\succeq \{n_1, n_3\} \sim \{n_1, n_4\} \sim \{n_2, n_3\} \sim \{n_2, n_4\} \succeq \{n_3\} \sim \{n_4\} \succeq \{n_1, n_2\} \succeq \{n_1\} \sim \{n_2\}$. In this case the solution is $\{n_3, n_4\}$, as it is the first sound (feasible) norm system in the ranking (on one hand, all norm systems containing 4 or 3 norms include a pair of exclusive norms and, on the other hand, out of all the norm systems with 2 norms it is the most preferred one).

Example 11 If $v_{fm} \sim_v v_{saf}$, when grounding these preferences with lex-cel we obtain the norm ranking $n_1 \sim_e n_2 \sim_e n_3 \sim_e n_4$. And lifting this ranking with anti-lex-cel we obtain: $\{n_1, n_2, n_3, n_4\} \succeq \{n_1, n_2, n_3\} \sim \{n_1, n_2, n_4\} \sim \{n_1, n_3, n_4\} \sim \{n_2, n_3, n_4\} \succeq \{n_1, n_2\} \sim \{n_1, n_3\} \sim \{n_1, n_4\} \sim \{n_2, n_3\} \sim \{n_2, n_4\} \sim \{n_3, n_4\} \succeq \{n_1\} \sim \{n_2\} \sim \{n_3\} \sim \{n_4\}$. In this case there are multiple solutions, namely: $\{n_1, n_2\}$, $\{n_1, n_4\}$, $\{n_2, n_3\}$, and $\{n_3, n_4\}$, as these are the most preferred sound (feasible) norm systems in the ranking (all norm systems containing 4 or 3 norms include a pair of exclusive norms, and the rest of norm systems with 2 norms are not sound). Notice that, when considering the values indifferently preferred, the possible solutions contain both permissive and restrictive norms for travellers. This was not the case in previous examples, as most permissive norms where selected when preferring freedom of movement over security and most restrictive norms when preferring security over freedom of movement.

10 Conclusions and future work

This paper describes tools to help decision makers select the most preferred set from a range of available options. We refer to this problem as the *dominant set selection problem* (DSSP). We solve it by considering both qualitative preference information over the features characterising the options and feasibility constraints.

Specifically, we propose to: first, ground the feature preferences to preferences over single objects; second, lift the object ranking into a ranking of all possible sets of options; and third, select the most preferred and feasible set of options. This requires the combination of existing results in the social choice literature, *lex-cel* [6], with our novel *anti-lex-cel*. To the best of our knowledge, the composition of ranking functions to obtain another ranking function (in our case we have composed *le* and *ale* to obtain *dom*) has not been previously explored and may pose an easier approach in those cases in which defining a ranking function directly is no easy task.

Moreover, we show how to encode the dominant set selection problem as a binary integer program (BIP) so that it can be solved with the aid of off-the-shelf solvers. We formally prove that solving the optimisation problem defined by the BIP encoding of the DSSP, we obtain a solution to the DSSP. Note that, for our set ranking we have found an encoding that allows to find the most preferred feasible set in the ranking while avoiding the computational cost of building it. We deem this strategy as promising when dealing with similar qualitative problems.

Thus, the overall contributions of this paper are two-fold. Firstly, the formalisation of a novel qualitative decision-making problem: the dominant set selection problem. Secondly, the resolution of this problem, which requires the combination of methods from the literature as well as our novel anti-lex-cel method. Regarding anti-lex-cel, we also characterise it axiomatically, prove its uniqueness and show that it generalises former results in

the literature. Furthermore, we use binary integer programming to encode (and solve) the problem. Finally, we illustrate our overall approach by means of the so-called value-alignment norm-selection problem.

However, our work counts on some limitations that are worth discussing. First, the qualitative approach described in this paper does not allow to express different degrees of relation with a feature (as it is the case in the quantitative approach described in [22]). Besides that, although we can express positive relationships between elements and features, we cannot express *negative* relationships (e.g. norm value demotion). Secondly, our work considers only element features, but cannot model set features. For instance, in a selection problem considering human candidates, gender is an individual feature. However, a feature such as *gender balance*, which affects a collection of individuals, cannot be handled within our framework.

As to future work, we plan to extend our work on norm selection. In its current form, we assume that norms promote moral values, but in general norms may also demote moral values. From a theoretical point of view, capturing demotion will require a reformulation of the dominant set selection problem. Also, we would like to study which problems cannot be efficiently solved with the current BIP encoding. Theoretically, we cannot efficiently solve any problem with constraints that are neither linear nor quadratic. In practice though, it is difficult to identify problems with such complex constraints. Finally, we also envision as an interesting future research to investigate alternative approaches to our dominance principle. Grounded on the literature of Multi-Criteria Decision Making, those approaches would compensate features rather than favouring element excellence.

Proofs

Axiomatic characterisation of anti-lex-cel

First, we prove that *ale* satisfies the axioms of our characterisation. After that, we show that, in fact, *ale* is the only lifting function that satisfies such axioms.

Proposition 1 *The anti-lex-cel lifting function ale satisfies Axioms* 1, 2, 3 *and* 4.

Proof 8 Let $\succeq_e$ be a ranking over the elements of X and $X/\sim_e$ its quotient order with equivalence classes $\Xi_1 \succ_e \ldots \succ_e \Xi_r$. Let $\succeq = ale(\succeq_e)$.

Axiom 1:

Take $x, y \in X$ such that $x \succeq_e y$ and $x \nsim_e y$. Then, two elements $i, j \in \{1, \ldots, r\}$ exist such that $i < j$, $x \in \Xi_i$ and $y \in \Xi_j$. So, $\eta(\{x\}) >_L \eta(\{y\})$ and, by relation (4), we have $\{x\} \succeq \{y\}$ and $\{x\} \nsim \{y\}$, which proves that *ale* satisfies Axiom 1.

Axiom 2:

let π be a bijection on X and let $\succeq_e^\pi \in \mathcal{R}(X)$ be such that

$$x \succeq_e y \Leftrightarrow \pi(x) \succeq_e^\pi \pi(y)$$

for all $x, y \in X$. Let $X/\sim_e^\pi$ be the quotient order of $\succeq_e^\pi$ with equivalence classes $\Xi_1^\pi \succ_e^\pi \ldots \succ_e^\pi \Xi_r^\pi$. Notice that $\succeq_e$ and $\succeq_e^\pi$ have precisely the same number of equivalence classes. Moreover, for any $x \in X$ there exists $i \in \{1, \ldots, r\}$ such that

$$x \in \Xi_i \Leftrightarrow \pi(x) \in \Xi_i^\pi.$$

So, $\eta(S) = \eta(\pi(S))$ for any $S \in \mathcal{P}(X)$ and where $\pi(S)$ is the image of S through π. By relation (4), it follows that *ale* also satisfies Axiom 2.

Axiom 3:

Consider two sets $S, S' \in \mathcal{P}(X)$ such that $S \succeq S'$. Consider the case $S \sim S'$ (the case $S \succeq S'$ and $S \nsim S'$ is similar and left to the reader). By relation (4) we have that $\eta(S) = \eta(S')$. Now take another set $\bar{S} \subseteq (X \setminus S)$, $\bar{S} \neq \emptyset$. Consider the new set $S \cup \bar{S}$ which contains some elements not in S. Then, $\eta(S \cup \bar{S}) >_L \eta(S) = \eta(S')$ and, by relation (4), it follows that $(S \cup \bar{S}) \succ S'$ and $(S \cup \bar{S}) \nsim S'$ which proves that *ale* satisfies Axiom 3.

Axiom 4:

Consider two sets $S, S' \in \mathcal{P}(X)$ such that $S \succeq S'$ and $S \nsim S'$. By relation (4), it exists $i \in \{1, \dots, r\}$ such that $\eta_k(S) = \eta_k(S)$ for all $k \in \{1, \dots, i-1\}$ and $\eta_i(S) > \eta_i(S')$ (being $\eta_j(S)$, the j-th element of $\eta(S)$). Let $\bar{S}' \subseteq (X \setminus S')$, $\bar{S}' \neq \emptyset$, be such that $x \succeq_e x'$ and $x \nsim_e x'$ for all $x \in S$ and $x' \in \bar{S}'$. Since each element in S is strictly preferred to each element in $\bar{S}'$, then $\eta_k(S') = \eta_k(S' \cup \bar{S}')$ for all $k \in \{1, \dots, i\}$ and, consequently, $\eta(S) >_L \eta(S' \cup \bar{S}')$, which finally proves the fact that *ale* also satisfies Axiom 4.

Having axiomatized anti-lex-cel, we now prove that anti-lex-cel is the only lifting function that satisfies these axioms. Such proof of unicity requires the previous introduction of two auxiliary lemmas.

Lemma 2 *Let $f : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ be a ranking lifting function that satisfies Axiom 2. Let $S, S' \in \mathcal{P}(X)$ and $\succeq_e \in \mathcal{R}(X)$ be such that $\eta(S) = \eta(S')$. Then $S \sim S'$ (where $\sim$ is the symmetric part of relation $\succeq = f(\succeq_e)$).*

Proof 9 Let $\succeq_e$ be a ranking over the elements of X and $X/\sim_e$ its quotient order with equivalence classes $\Xi_1 >_e \dots >_e \Xi_r$. Since $\eta(S) = \eta(S') = (s_1, \dots, s_r)$, with $s_i = |S \cap \Xi_i| = |S' \cap \Xi_i|$ for all $i \in \{1, \dots, r\}$, we can define a bijection π on X such that the $\pi(S \cap \Xi_i) = S' \cap \Xi_i$ and $\pi(S' \cap \Xi_i) = S \cap \Xi_i$ for all $i \in \{1, \dots, r\}$. So, $\pi(S) = S'$ and $\pi(S') = S$.

Define a new ranking $\succeq_e^\pi \in \mathcal{R}(X)$ such that $x \succeq_e y \Leftrightarrow \pi(y) \succeq_e^\pi \pi(y)$ for all $x, y \in X$. Then, by Axiom 2, we have that

$$S \succeq S' \Leftrightarrow S' \succeq^\pi S.$$

On the other hand, $\succeq = \succeq^\pi$, and we may conclude that $S \succeq S' \Leftrightarrow S' \succeq S$, which precisely means that $S \sim S'$ for $\succeq$ is a total relation.

Lemma 3 *Given an element ranking $\succeq_e \in \mathcal{R}(X)$ and $f : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ a ranking lifting function that satisfies Axioms 1 and 4. Then, the resulting set ranking $f(\succeq_e) = \succeq$ is such that $\forall x \in X, \{x\} \succeq S$ and $\{x\} \nsim S$ for every $S \subseteq L(x)$, where $L(x) = \{x' \in X | x \succeq_e x'$ and $x \nsim_e x'\}$, is the set of elements strictly less preferred than x.*

Proof 10 Given an element ranking $\succeq_e \in \mathcal{R}(X)$, let $\succeq = f(\succeq_e)$ and $\succeq^* = ale(\succeq_e)$. Let $x \in X$ and $L(x) = \{x' \in X | x \succeq_e x'$ and $x \nsim_e x'\}$. Take $y \in L(x)$. By Axiom 1, we have that $\{x\} \succeq \{y\}$ and $\{x\} \nsim \{y\}$. Now take $\bar{S}' \subseteq L(x) \setminus \{y\}$. By Axiom 4, with $\{x\}$ in the role of S, $\{y\}$ in the role of S', we have that $\{x\} \succeq \{y\} \cup \bar{S}'$ and $\{x\} \nsim \{y\} \cup \bar{S}'$. Since $\{y\} \cup \bar{S}'$ can be whatever subset of $L(x)$, we have proved the lemma.

Theorem 1 *Let $f : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ be a ranking lifting function. Then f satisfies Axioms* 1, 2, 3 *and* 4 *if and only if f is the anti-lex-cel lifting function ale.*

Proof 11 We know from Proposition 1 that the anti-lex-cel lifting function *ale* satisfies Axioms 1, 2, 3 and 4.

Conversely, suppose f satisfies Axioms 1, 2, 3 and 4. Consider an element ranking $\succeq_e \in \mathcal{R}(X)$ having $X/ \sim_e$ as quotient order with equivalence classes $\Xi_1 >_e \ldots >_e \Xi_r$. Let $\succeq = f(\succeq_e)$ and $\succeq^* = ale(\succeq_e)$. It has to be proved that

$$Q \succeq^* Q' \Leftrightarrow Q \succeq Q'$$

for all $Q, Q' \in \mathcal{P}(X)$.

We first prove the equivalence for the asymmetric parts, i.e.,

$$Q \succeq^* Q' \text{ and } Q \nsim^* Q' \Leftrightarrow Q \succeq Q' \text{ and } Q \nsim Q'$$

for all $Q, Q' \in \mathcal{P}(X)$.

($\Rightarrow$)

Let $Q \succeq^* Q'$ and $Q \nsim^* Q'$. By relation (4), it exists $i \in \{1, \ldots, r\}$ such that $\eta_k(Q) = \eta_k(Q')$ for all $k \in \{1, \ldots, i-1\}$ and $\eta_i(Q) > \eta_i(Q')$ (being $\eta_j(Q)$, the j-th element of $\eta(Q)$). We distinguish two cases:

(i): $\eta_k(Q') = 0$ for all $k \in \{1, \ldots, i\}$. Take an element $x \in \Xi_i \cap Q$ and an element $y \in Q' \cap \Xi_j$ where $j \in \{i+1, \ldots, r\}$ is the smallest index such that $Q' \cap \Xi_j \neq \emptyset$. Notice that $j > i$, so $x \succeq_e y$ and $x \nsim_e y$ for all $y \in Q'$. Since f satisfies both Axiom 1 and 4, by Lemma 3 we have that $\{x\} \succeq Q'$ and $\{x\} \nsim Q'$. Now, let $\bar{Q} = Q \setminus \{x\}$. Being $Q = \{x\} \cup \bar{Q}$, and applying Axiom 3 on f with $\{x\}$ in the role of S, Q' in the role of S' and $\bar{Q}$ in the role of $\bar{S}$, we finally have $Q \succeq Q'$ and $Q \nsim Q'$.

(ii): $\eta_k(Q') \neq 0$ for some $k \in \{1, \ldots, i\}$. First, consider the two non-empty sets $T = \bigcup_{k \in \{1, \ldots, i\}} (Q \cap \Xi_k)$ and $T' = \bigcup_{k \in \{1, \ldots, i\}} (Q' \cap \Xi_k)$. Since $\eta_i(Q) = |Q \cap \Xi_i| > |Q' \cap \Xi_i| = \eta_i(Q')$ there must be at least $\eta_i(Q) - \eta_i(Q')$ elements in $Q \cap \Xi_i$ but not in $Q' \cap \Xi_i$. Let $I = \{x \in Q \cap \Xi_i \setminus Q'\}$ with $|I| = \eta_i(Q) - \eta_i(Q')$. Now, consider the two sets $T \setminus I$ and T'. By construction, $\eta(T \setminus I) = \eta(T')$. Then, since f satisfies Axiom 2, by Lemma 2 we have that $T \setminus I \sim T'$. We are now ready to apply Axiom 3 on f with $T \setminus I$ in the role of S and T' in the role of S' and $\bar{T} = Q \setminus (T \setminus I)$ in the role of $\bar{S}$. Then, being $Q = (T \setminus I) \cup \bar{T}$, we have $Q \succeq T'$ and $Q \nsim T'$. Finally, we use Axiom 4 on f with Q in the role of S, T' in the role of S' and $\bar{T}' = Q' \setminus T'$ in the role of $\bar{S}'$. So, we have that $Q \succeq T' \cup \bar{T}'$ and $Q \nsim T' \cup \bar{T}'$. Being $Q' = T' \cup \bar{T}'$, we conclude that $Q \succeq Q'$ and $Q \nsim Q'$.

($\Leftarrow$)

Let $Q \succeq Q'$ and $Q \nsim Q'$. First, suppose that $Q \sim^* Q'$. Then, by relation (4), we have that $\eta(Q) = \eta(Q')$. So, by Lemma 2, it must be $Q \sim Q'$, which yields a contradiction. On the other hand it cannot even be $Q' \succeq^* Q$ and $Q' \nsim^* Q$ (otherwise we would have a contradiction by the other implication proved earlier). So, by the fact that $\succeq^*$ is a total relation, it must be $Q \succeq^* Q'$ and $Q \nsim^* Q'$, which concludes the proof of the equivalence between the asymmetric parts of $\succeq$ and $\succeq^*$.

We now prove the equivalence for the symmetric parts, i.e.,

$$Q \sim^* Q' \Leftrightarrow Q \sim Q'$$

for all $Q, Q' \in \mathcal{P}(X)$.

($\Rightarrow$)

Let $Q \sim^* Q'$. Then, by relation (4), we have that $\eta(Q) = \eta(Q')$ and by Lemma 2 and the fact that $\succeq$ satisfies Axiom 2, it immediately follows that $Q \sim Q'$.

($\Leftarrow$)

Let $Q \sim Q'$. By the equivalence of the asymmetric part proved earlier, we cannot have $Q \succeq^* Q'$ and $Q \nsim^* Q'$, or, $Q' \succeq^* Q$ and $Q' \nsim^* Q$. So, since $\succeq^*$ is a total relation, it must be $Q \sim^* Q'$, which concludes the proof.

In the following, we prove that the four axioms 1, 2, 3 and 4 are logically independent, so they all are necessary to uniquely characterize *ale*.

Proposition 3 *Exists a function $f \neq ale$ satisfying any three of the axioms 1, 2, 3 and 4 and not fulfiling the remaining one.*

Proof 12 *A ranking lifting function that does not satisfy simple dominance.* Consider a ranking lifting function $f^{sd} : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ with $\succeq = f^{sd}(\succeq_e)$ such that $S \succeq T \Leftrightarrow \eta(S) \geq_L \eta(T)$ for all $S, T \in \mathcal{P}(X)$ with $|S| > 1$ or $|T| > 1$, and $\{x\} \sim \{y\}$ for all $x, y \in X$. Similar to *ale*, it is easy to verify along the lines of Proposition 1 that f^{sd} satisfies axioms 2, 3 and 4. But of course f^{sd} does not satisfies Axiom 1 (just take $\succeq_e$ such that $x \succeq_e y$ and $x \nsim_e y$).

A ranking lifting function that does not satisfy neutrality. Let $z \in X$. Consider a ranking lifting function $f^n : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ with $\succeq = f^n(\succeq_e)$ such that $S \succeq T$ and $S \nsim T$ for all $S, T \in \mathcal{P}(X)$ with $\eta(S) = \eta(T)$ and $z \in S \setminus T$, and $S \succeq T \Leftrightarrow \eta(S) \geq_L \eta(T)$ for all the remaining pairs of sets $S, T \in \mathcal{P}(X)$. It easy to verify along the lines of Proposition 1 that f^n satisfies axioms 1, 3 and 4. But of course f^n does not satisfies Axiom 2, for f^n breaks some ties in *ale* if favour of the set containing element z.

A ranking lifting function that does not satisfy size monotonicity. Consider a ranking lifting function $f^{sm} : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ with $\succeq = f^{sm}(\succeq_e)$ such that $S \succeq T \Leftrightarrow i_S \leq i_T$ for all $S, T \in \mathcal{P}(X)$, where i_S and i_T are, respectively, the smallest index such that $\eta_{i_S}(S) \neq 0$ and $\eta_{i_T}(T) \neq 0$. One can check that f^{sm} satisfies Axioms 1, 2 and 4, but it does not fulfil Axiom 3. To see that f^{sm} does not satisfy Axiom 3, consider $X = \{x_1, x_2, x_3\}$ with the element ranking $x_1 \succeq_e x_2 \succeq_e x_3$, which implies $\eta(\{x_1\}) = (1, 0, 0)$, $\eta(\{x_2\}) = (0, 1, 0)$ and $\eta(\{x_3\}) = (0, 0, 1)$. Then, $\{x_1, x_2\} \sim \{x_1, x_3\}$ (for $\eta_1(\{x_1, x_2\}) = \eta_1(\{x_1, x_3\}) = 1 \neq 0$), but it is not true that $\{x_1, x_2\} \cup \{x_3\} \succeq \{x_1, x_3\}$ and $\{x_1, x_2\} \cup \{x_3\} \nsim \{x_1, x_3\}$ (for $\eta_1(\{x_1, x_2, x_3\}) = \eta_1(\{x_1, x_3\}) = 1 \neq 0$).

A ranking lifting function that does not satisfy independence of the worst elements. Consider a ranking lifting function $f^{iwe} : \mathcal{R}(X) \to \mathcal{R}(\mathcal{P}(X))$ with $\succeq = f^{iwe}(\succeq_e)$ and such that $S \succeq T \Leftrightarrow |S| \geq |T|$ for all $S, T \in \mathcal{P}(X)$ with $|S| > 1$ or $|T| > 1$, and $\{x\} \succeq \{y\} \Leftrightarrow x \succeq_e y$ for all $x, y \in X$. One can check that f^{iwe} satisfies Axioms 1, 2 and 3, but it does not fulfil Axiom 4. To see that f^{iwe} does not satisfy Axiom 4, consider $X = \{x_1, x_2, x_3, x_4\}$ and the element ranking $x_1 \sim_e x_2 \succeq_e x_3 \sim_e x_4$ with quotient order $\Xi_1 = \{x_1, x_2\} \succ_e \Xi_2 = \{x_3, x_4\}$. We have that $\{x_1, x_2\} \succeq \{x_2\}$ and $\{x_1, x_2\} \nsim \{x_2\}$ (for $|\{x_1, x_2\}| = 2$ and $|\{x_2\}| = 1$) but it is not true that $\{x_1, x_2\} \succeq \{x_2\} \cup \{x_3, x_4\}$ and $\{x_1, x_2\} \nsim \{x_2\} \cup \{x_3, x_4\}$ (for $|\{x_2, x_3, x_4\}| = 3$).

$\succeq$ embodies dominance

To show that $\succeq$ embodies dominance, first we need an auxiliary result showing that $\succeq_e$ embodies element dominance.

Lemma 4 $\succeq_e = le(\succeq_F)$ *embodies element dominance, that is* $\forall x, y \in X$, x *is dominant over* y $\Leftrightarrow x \succ_e y$.

Proof 13 To start we have a set of features F and a ranking over them $\succeq_F$, which can be represented in general as: $f_1^1 \sim_F \cdots \sim_F f_1^s \succ_F \cdots \succ_F f_k^1 \sim_F \cdots \sim_F f_k^r$ meaning that in $F/\sim_F$ the quotient order is $\Psi_1 \succ_F \cdots \succ_F \Psi_k$, with $\Psi_i = \{f_i^1, \dots f_i^q\}$($q$ depends on i, for Ψ_1, $q = s$ and for Ψ_k, $q = r$).

With these considerations, suppose $x \succ_e y$, by the definition of lex-cel, this means that $\mu(x) >_L \mu(y)$, which by the definition of μ means that $\exists \Psi_i$, such that x has a larger number of features in Ψ_i than y: $|\mathfrak{f}(x) \cap \Psi_i| > |\mathfrak{f}(y) \cap \Psi_i|$, while $\forall \Psi_j \succ_F \Psi_i$, x and y have the same number of features in Ψ_j: $|\mathfrak{f}(x) \cap \Psi_j| = |\mathfrak{f}(y) \cap \Psi_j|$. This means that x is Ψ_i-dominant over y, while for all $\Psi_j \succ_F \Psi_i$, they are Ψ_j-indifferent, which is the definition of x being dominant over y.

Now having proved that $x \succ_e y$ implies x dominant over y, we tackle the other direction. Suppose x dominant over y, if $x \prec_e y$ we have seen that would imply y dominant over x which contradicts our assumption, therefore $x \succeq_e y$, but note that if $x \sim_e y$, then proceeding as above we obtain $|\mathfrak{f}(x) \cap \Psi_i| = |\mathfrak{f}(y) \cap \Psi_i| \forall \Psi_i$, which would mean that neither x dominates y nor vice versa, contradicting our initial assumption. Therefore the only possibility is that $x \succ_e y$, proving the lemma.

Having seen that $\succeq_e$ embodies element dominance, we can build upon this result to prove the theorem stating that $\succeq$ embodies dominance.

Theorem 2 *Let X be a set of elements, F a set of features, $\succeq_F$ a ranking over F and $\mathfrak{f}$ a function relating elements to their features, then for $S, S' \in \mathcal{P}(X)$, S is dominant over $S' \Leftrightarrow S \succ S'$ ($S \succeq S'$ and $S \nsim S'$), where $dom(\succeq_F) = ale(le(\succeq_F)) = \succeq$.*

Proof 14 First, suppose $S \succ S'$, then since $ale(le(\succeq_F)) = \succeq$ we know that $\eta(S) >_L \eta(S')$, that is $\exists \Xi_i \in X/\sim_e$, such that $|S \cap \Xi_i| > |S' \cap \Xi_i|$ and $\forall \Xi_j \in X/\sim_e$, $\Xi_j \succ_e \Xi_i$, $|S \cap \Xi_j| = |S' \cap \Xi_j|$, note though that these equivalence classes are ordered with $\succ_e$ which we have seen in the lemma that embodies element dominance, therefore for $\Xi \succ_e \Xi'$, all elements of Ξ are dominant over all elements of Ξ', while the elements in the equivalence class are indifferent between them. With this consideration and the previous findings $|S \cap \Xi_i| > |S' \cap \Xi_i|$ and $\forall \Xi_j \in X/\sim_e$, $\Xi_j \succ_e \Xi_i$, $|S \cap \Xi_j| = |S' \cap \Xi_j|$ means that S contains more elements in Ξ_i than S', while containing the same number of elements for more preferred equivalence classes. Therefore, considering $S = \{s_1, \dots, s_{|S|}\}$, $S' = \{s'_1, \dots, s'_{|S'|}\}$ and the permutation σ explained in Sect. 4 (and used in the definition of dominance), we have that each $s_{\sigma(1)}, \dots, s_{\sigma(r)}$ is indifferent with its counterpart $s'_{\sigma(1)}, \dots, s'_{\sigma(r)}$ for $r = \sum_{j<i} |S \cap \Xi_j| + |S' \cap \Xi_i|$, but $s_{\sigma(r+1)}$ dominates $s'_{\sigma(r+1)}$, because either $s_{\sigma(r+1)} \in \Xi_i$ and $s'_{\sigma(r+1)} \in \Xi_l$ with $l > i$ or $r + 1 > |S'|$, which is the definition of S being dominant over S'.

Now having proved that $S \succ S'$ implies S dominant over S', we tackle the other implication. Suppose S dominant over S', if $S \prec S'$ we have seen that would imply S' dominant over S which contradicts our assumption, therefore $S \succeq S'$, but note that if $S \sim S'$, then

$|S \cap \Xi| = |S' \cap \Xi| \forall \Xi \in X/\sim_e$ and therefore $s_{\sigma(i)}$ and $s'_{\sigma(i)}$ are in the same equivalence class $\forall i$, meaning that $s_{\sigma(i)}$ and $s'_{\sigma(i)}$ are indifferent $\forall i$, which means that S and S' are indifferent contradicting our assumption. Therefore the only possibility is that $S \succ S'$, which proves the theorem.

$\mathfrak{p}$ embodies the $\succeq$ ranking

To show the main goal of this section, namely that $\mathfrak{p}$ embodies the $\succeq$ ranking, we firstly require some lemmas regarding the properties of $\mathfrak{p}$.

Lemma 5 $\mathfrak{p}(S) = \sum_{w=1}^{r} \mathfrak{p}(S \cap \Xi_w)$

Proof 15 By applying Eq. 5 we obtain the preference of an equivalence class Ξ_w as $\mathfrak{p}(S \cap \Xi_w) = \sum_{i=1}^{r} |S \cap \Xi_w \cap \Xi_i|(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1) = |S \cap \Xi_w|(\sum_{j=w+1}^{r} \mathfrak{p}(\Xi_j) + 1)$, since all equivalence classes are disjoint, meaning that $|S \cap \Xi_w \cap \Xi_i| = |\emptyset| = 0$ when $i \neq w$ and $|S \cap \Xi_w \cap \Xi_i| = |S \cap \Xi_w|$, when $i = w$. Now $\sum_{w=1}^{r} \mathfrak{p}(S \cap \Xi_w) = \sum_{w=1}^{r} \sum_{i=1}^{r} |S \cap \Xi_w \cap \Xi_i|(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1) = \sum_{w=1}^{r} |S \cap \Xi_w|(\sum_{j=w+1}^{r} \mathfrak{p}(\Xi_j) + 1) = \mathfrak{p}(S)$.

Lemma 6 $\forall w, \mathfrak{p}(\Xi_w) \geq \mathfrak{p}(S \cap \Xi_w)$

Proof 16 Since all equivalence classes are disjoint, from Eq. 5 we have that $\mathfrak{p}(\Xi_w) = |\Xi_w|(\sum_{j=w+1}^{r} \mathfrak{p}(\Xi_j) + 1)$ and $\mathfrak{p}(S \cap \Xi_w) = |S \cap \Xi_w|(\sum_{j=w+1}^{r} \mathfrak{p}(\Xi_j) + 1)$. Since $|\Xi_w| \geq |S \cap \Xi_w|$, then $\mathfrak{p}(\Xi_w) \geq \mathfrak{p}(S \cap \Xi_w)$.

With these lemmas in mind, we now prove that $\mathfrak{p}$ embodies the $\succeq$ ranking by means of the following theorem.

Theorem 3 *Given two sets $S, S' \in \mathcal{P}(X)$, $S \succeq S' \Leftrightarrow \mathfrak{p}(S) \geq \mathfrak{p}(S')$.*

Proof 17 We divide the proof into three steps. First we prove two implications, and we subsequently show that these implications suffice to prove the theorem.

$S \succ S' \Rightarrow \mathfrak{p}(S) > \mathfrak{p}(S')$: Say that $S \succ S'$. From Eq. 4, we have that $S \succ S' \Leftrightarrow \eta(S) >_L \eta(S')$. By using the definition of η in Eq. 3 we can write $\eta(S) >_L \eta(S')$ as $(c_1^S, \ldots c_r^S) >_L (c_1^{S'}, \ldots c_r^{S'})$ (where $c_i^S = |S \cap \Xi_i|$ and $c_i^{S'} = |S' \cap \Xi_i| \forall i$). Now, by using the formalisation of the lexicographical order (see Definition 5), we have that $(c_1^S, \ldots c_r^S) >_L (c_1^{S'}, \ldots c_r^{S'})$, which implies that $\exists k \in \{1, \ldots, r\}$, s.t. $\forall t < k, c_t^S = c_t^{S'}$ and $c_k^S > c_k^{S'}$. In other words, $\exists k \in \{1, \ldots, r\}$ s.t. $|S \cap \Xi_k| > |S' \cap \Xi_k|$ and $\forall t < k, |S \cap \Xi_t| = |S' \cap \Xi_t|$ and therefore $\mathfrak{p}(S \cap \Xi_t) = \mathfrak{p}(S' \cap \Xi_t)$.

Next we prove that $\mathfrak{p}(S) > \mathfrak{p}(S')$. First, note that by considering Lemma 5, we have that $\mathfrak{p}(S) = \sum_{i=1}^{k-1} \mathfrak{p}(S \cap \Xi_i) + \sum_{i=k}^{r} \mathfrak{p}(S \cap \Xi_i) \geq \sum_{i=1}^{k-1} \mathfrak{p}(S \cap \Xi_i) + \mathfrak{p}(S \cap \Xi_k)$ and applying Lemma 5 and Lemma 6 we have that $\mathfrak{p}(S') = \sum_{i=1}^{k-1} \mathfrak{p}(S' \cap \Xi_i) + \sum_{i=k}^{r} \mathfrak{p}(S' \cap \Xi_i) \leq \sum_{i=1}^{k-1} \mathfrak{p}(S' \cap \Xi_i) + \mathfrak{p}(S' \cap \Xi_k) + \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i)$. Therefore, to prove that $\mathfrak{p}(S) > \mathfrak{p}(S')$ it suffices to prove that $\sum_{i=1}^{k-1} \mathfrak{p}(S \cap \Xi_i) + \mathfrak{p}(S \cap \Xi_k) > \sum_{i=1}^{k-1} \mathfrak{p}(S' \cap \Xi_i) + \mathfrak{p}(S' \cap \Xi_k) + \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i)$. This is equivalent to show that $\mathfrak{p}(S \cap \Xi_k) - \mathfrak{p}(S' \cap \Xi_k) - \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i) > 0$.

Now, using Eq. 5, $\mathfrak{p}(S \cap \Xi_k) - \mathfrak{p}(S' \cap \Xi_k) - \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i) = |S \cap \Xi_k|(\sum_{j=k+1}^{r} \mathfrak{p}(\Xi_j) + 1) - |S' \cap \Xi_k|(\sum_{j=k+1}^{r} \mathfrak{p}(\Xi_j) + 1) - \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i) = (|S \cap \Xi_k| - |S' \cap \Xi_k|)(\sum_{j=k+1}^{r} \mathfrak{p}(\Xi_j) + 1) - \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i)$.

As shown above, we know that $|S \cap \Xi_k| > |S' \cap \Xi_k|$. From that, and since these sets' cardinalities are natural numbers, we obtain the following lower bound: $|S \cap \Xi_k| - |S' \cap \Xi_k| \geq 1$. Therefore, $(|S \cap \Xi_k| - |S' \cap \Xi_k|)(\sum_{j=k+1}^{r} \mathfrak{p}(\Xi_j) + 1) - \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i)) \geq \sum_{j=k+1}^{r} \mathfrak{p}(\Xi_j) + 1 - \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i)) = 1 > 0$.

Recall that we assumed that $S \succ S'$. Since we have managed to prove that $S \succ S'$ implies that $\mathfrak{p}(S \cap \Xi_k) - \mathfrak{p}(S' \cap \Xi_k) - \sum_{i=k+1}^{r} \mathfrak{p}(\Xi_i) > 0$, which in turn implies that $\mathfrak{p}(S) > \mathfrak{p}(S')$, then it is clear that $S \succ S' \Rightarrow \mathfrak{p}(S) > \mathfrak{p}(S')$.

$S \succ S' \Leftarrow \mathfrak{p}(S) > \mathfrak{p}(S')$: Suppose that $\mathfrak{p}(S) > \mathfrak{p}(S')$. If $S \prec S'$, then we have already shown above that $\mathfrak{p}(S) < \mathfrak{p}(S')$, which contradicts our initial assumption. If $S \sim S'$, then $\eta(S) = \eta(S')$, which means that $(c_1^S, \ldots, c_r^S) = (c_1^{S'}, \ldots, c_r^{S'})$, and therefore $\forall i \; c_i^S = c_i^{S'}$. This means that $\forall i \, |S \cap \Xi_i| = |S' \cap \Xi_i|$, which implies that $\mathfrak{p}(S) = \sum_{i=1}^{r} |S \cap \Xi_i|(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1)$ $= \sum_{i=1}^{r} |S' \cap \Xi_i|(\sum_{j=i+1}^{r} \mathfrak{p}(\Xi_j) + 1) = \mathfrak{p}(S')$. The fact that $\mathfrak{p}(S) = \mathfrak{p}(S')$ also contradicts our initial assumption $\mathfrak{p}(S) > \mathfrak{p}(S')$. Thus, we conclude that $\mathfrak{p}(S) > \mathfrak{p}(S') \Rightarrow S \succ S'$.

$S \succ S' \Leftrightarrow \mathfrak{p}(S) > \mathfrak{p}(S')$ suffices to prove the theorem: Note that we have proved that $S \succ S' \Leftrightarrow \mathfrak{p}(S) > \mathfrak{p}(S')$, then it trivially follows that $S \prec S' \Leftrightarrow \mathfrak{p}(S) < \mathfrak{p}(S')$. And these two cases imply that $S \sim S' \Leftrightarrow \mathfrak{p}(S) = \mathfrak{p}(S')$. Finally, $S \succ S' \Leftrightarrow \mathfrak{p}(S) > \mathfrak{p}(S')$ and $S \sim S' \Leftrightarrow \mathfrak{p}(S) = \mathfrak{p}(S')$ imply that $S \succeq S' \Leftrightarrow \mathfrak{p}(S) \geq \mathfrak{p}(S')$, which ends the proof of the theorem.

DSSP algorithm and implementation

Algorithm 1 encodes a DSSP into a BIP, and also writes this encoding into a file that can be fed into a BIP solver. This algorithm receives as input:

- a non-empty list of elements X;
- a list F of feature equivalence classes in descending order of preference (each equivalence class being a list of its indifferently preferred features, hence the feature order $f1 \succ_F f2 \sim_F f3 \succ_F f4$ would be represented as as $F = [[f1], [f2, f3], [f4]]$);
- a mapping $\mathfrak{f}$ relating elements to their features; and
- a list of constraints C (each constraint $c \in C$ being a string).

The algorithm uses several auxiliary functions: sort(l, k) sorts a list l in ascending order using as key a function k; write(s, file) writes a string s in a separated line in the given file; str(num) converts a numeric value num into a string; and get(l, i), which returns the element in position i in a list l. Finally, given two strings s and s', we represent string concatenation as $s + s'$. Notice also that "+" represents a string solely composed by character '+'.

Algorithm 2 provides a function used in Algorithm 1 to compute profile vectors according to the definition in Sect. 6. Algorithm 2 receives as input an element $x \in X$, a list F of feature equivalence classes in descending order of preference, and a mapping $\mathfrak{f}$ relating elements to their features. From that, it builds the profile of the element, $\mu(x)$ (as a list instead of a vector), following Eq. 1. The auxiliary function has_feature in Algorithm 2 checks if an element has a given feature or not.

There is a publicly-available implementation of our DSSP encoder and solver at https://gitlab.iiia.csic.es/marcserr/dssp.

Example 12 below shows an example of the BIP output by Algorithm 1.

Algorithm 1 Encoding a DSSP into a BIP

1: **procedure** BUILD_BIP(X, F, ƒ, C)
2:    FILE bip   ▷ Create an empty BIP file
3:    pref ← 1   ▷ Initialise preference variable
4:    toadd ← 1   ▷ This variable accumulates preferences while "traversing" the same
      equivalence class, and it is added to pref when changing to another equivalence class
5:    i ← 1   ▷ Initialise index
6:    sort(X, MU(x, F, ƒ)) ▷ Lex-cel uses μ in Alg.2 to sort X from least to most preferred
7:    write("Maximize", bip)   ▷ Write objective function
8:    objfunc ← str(pref)+get(X, i)   ▷ Initialise objective function to "$1x_1$"
9:    **while** $i \leq |X|$ **do**
10:       i ← i + 1   ▷ Increase index i
11:       **if** MU(get(X,i),F, ƒ) = MU(get(X,i-1),F, ƒ) **then** ▷ $x_i \sim_e x_{i-1}$ in the same class
12:          objfunc ← objfunc + "+" + str(pref) + get(X, i) ▷ eg: objfunc="$1x_1 + 1x_2$"
13:          toadd ← toadd + pref
14:       **else**   ▷ $x_i \succ_e x_{i-1}$ in a more preferred equivalence class
15:          pref ← pref + toadd
16:          objfunc ← objfunc + "+" + str(pref) + get(X, i) ▷ eg.: objfunc="$1x_1 + 2x_2$"
17:          toadd ← pref
18:       **end if**
19:    **end while**
20:    write(objfunc, bip)
21:    write("Subject To", bip)   ▷ Write constraints
22:    **for** $c \in C$ **do**
23:       write(c, bip)
24:    **end for**
25:    write("Binaries", bip)   ▷ Write binary decision variables
26:    **for** $x \in X$ **do**
27:       write(x, bip)
28:    **end for**
29:    write("End", bip)
30: **end procedure**

Algorithm 2 Compute profile vector μ (see Section 6)

1: **procedure** MU(x, F, ƒ)
2:    mu ← An empty list   ▷ Initialise profile vector
3:    **for** $c \in$ F **do**   ▷ From most preferred to least preferred feature equivalence class
4:       $num \leftarrow 0$ ▷ Counter of features of x in class c
5:       **for** $f \in c$ **do**   ▷ For all features in the equivalence class
6:          **if** has_feature(x, f, ƒ) **then**
7:             $num \leftarrow num + 1$ ▷ Increase feature counter
8:          **end if**
9:       **end for**
10:       append(num, mu)   ▷ Append num at the end of profile vector
11:    **end for**
12:    **return** mu
13: **end procedure**

Example 12 Suppose that a school wants to award three scholarships to their best last year students. Eight candidate students are considered, for which we assign decision variables $X = \{s1, s2, \ldots, s8\}$ (where $si = 1$ means student si is awarded with a scholarship and $si = 0$ means that si is not). The features that the school considers are: academic excellence (ax), good behaviour (gb), having helped the staff (hl), and punctuality (p). Academic excellence is the most important feature to compete for a scholarship, followed by good behaviour and helping staff members, which are indifferently preferred. Finally, punctuality is the least preferred feature. Hence, the feature order is: ax ≻ gb ∼ hl ≻ p, meaning that F= [[ax], [gb, hl], [p]]. The features of each student are:

$$\text{s1: ax, gb, hl s2: ax, gb, hl, p s3: gb, hl, p s4: ax, gb, p}$$
$$\text{s5: gb, hl, p s6: hl, p s7: ax, gb, p s8: gb, hl}$$

Finally, in terms of constraints, since there are only three scholarships, we have to consider a constraint that enforces that we have to exactly select three students: s1+s2+s3+s4+s5+s6+s7+s8=3. Furthermore, suppose that s1-s4 belong to one group, whereas s5-s8 belong to another group, and the school wants to give at least one scholarship to each group. Thus, we have to consider constraints "s1+s2+s3+s4>= 1" and "s5+s6+s7+s8>= 1". Then, the resulting BIP after applying Algorithm 1 would be :

Maximize
1s6 + 2s8 + 4s3 + 4s5 + 12s4 + 12s7 + 36s1 + 72s2
Subject To
s1+s2+s3+s4+s5+s6+s7+s8=3
s1+s2+s3+s4>= 1
s5+s6+s7+s8>= 1
Binaries
s1
s2
s3
s4
s5
s6
s7
s8
End

The solution to the BIP above is {s1, s2, s7}.

VANS algorithm and implementation

Algorithm 3 encodes a VANS problem into a BIP. The input of the algorithm contains: a non-empty set of norms N; a list of value equivalence classes V (each equivalence class being a list of its indifferently preferred values); a mapping relating norms to their promoted values ƒ; a list of mutually exclusive relations (the relations being binary tuples); and a generalisation graph (implementing generalisation relations, being the parents of a norm, more general norms). First, the algorithm builds the constraints on norms discussed in Sect. 9 using Algorithm 4. Thereafter, it feeds *N*, *V*, ƒ, and the obtained constraints into Algorithm 1 in Appendix B.

Algorithm 4 uses several auxiliary functions (some of them already used in Appendix B): append(x, list) appends x at the end of the list; get(l, i) returns the element in position i of the list l; parents(n, G) returns a list of all parents (direct or not) of n in the graph G; direct_siblings(n, G) returns a list of the direct siblings of n in G. Finally, given two strings s and s', we represent string concatenation as s + s'. Notice also that "+" represents a string solely composed by character '+'.

Example 13 below shows the BIP obtained after applying Algorithm 3 to Example 8.

Example 13

- BIP file:
- Maximize
- 3n1 + 3n2 + 1n3 + 1n4
- Subject To
- n1 + n3 <= 1
- n2 + n4 <= 1
- Binaries
- n1
- n2
- n3
- n4
- End

The solution to the BIP above is {n1, n2}.

There is a publicly-available implementation of our encoder for VANS problems at https://gitlab.iiia.csic.es/marcserr/vans-problem.

Algorithm 3 Encoding a VANS into a BIP

---

procedure VANS_BIP(N, V, $\mathfrak{f}$, R_x, G)
    C ← VANS_CONSTRAINTS(N, R_x, R_g)   ▷ See Algorithm 4
    BUILD_BIP(N, V, $\mathfrak{f}$, C)   ▷ See Algorithm 1 in appendix B
end procedure

Algorithm 4 Building constraints for VANS from norm relations

---

procedure VANS_CONSTRAINTS(N, R_x, G)
    C ← An empty list   ▷ Initialise constraints
    for $(n1, n2) \in R_x$ **do**   ▷ Loop over norm exclusivity relations
        excon ← n1 + " + " + n2 + " <= 1"   ▷ Exclusivity constraint as in Eq. 8
        append(excon, C)
    end for
    for $n \in$ N **do**
        for p $\in$ parents(n, G) **do**   ▷ Look at all the parents of norm n
            gencon ← n + " + " + p + " <= 1"   ▷ Generate constraints based on Eq. 9
            append(gencon, C)
        end for
        if |direct_siblings(n, G)| > 1 **then**
            i ← 1
            gencon ← get(direct_siblings(n, G), i) ▷ Generate constraints based on Eq. 10
            while i $\leq$ |direct_siblings(n, G)| **do**
                i ← i + 1
                gencon ← gencon + " + " + get(direct_siblings(n, G), i)
            end while
            gencon ← gencon + " <= " + str(|direct_siblings(n, G)| - 1)
            append(gencon, C)
        end if
    end for
    return C
end procedure

List of notation and symbols

General notation

X	A set of elements.
$x/x'/x_i$	An element $x/x'/x_i \in X$.
$\mathcal{P}(X)$	The power set of X.
$S/S'/S_i$	A subset of elements of X, $S/S'/S_i \in \mathcal{P}(X)$.
$\geq/\geq'/\geq_X$	A generic ranking, where $>/>'/>_X$ is its antisymmetric part and $\sim/\sim'/\sim_X$ its symmetric part. We also use this notation for the ranking over $\mathcal{P}(X)$ obtained through anti-lex-cel.
F	A set of features.
$\mathcal{R}(X)$	All possible rankings over a set.
$\geq_F$	A ranking of features.
ƒ	A function that receives and element in X and returns the set of its features.
ϕ	The feasibility function that receives a set in $S \in P(X)$ and returns $\top$ if the set is feasible and $\bot$ if it is not feasible.
$\Psi/\Psi'/\Psi_i$	A feature equivalence class in $F/\sim_F$.
σ	A permutation of indexes of the elements in a set S with respect to a property (dominance). Thus, $\sigma(i)$ is the index of the i-th best element with regards to the property.
srs	A social ranking solution, that is a function $srs : \mathcal{R}(\mathcal{P}(X)) \rightarrow \mathcal{R}(X)$.
le	The lex-cel (lexicographic excellence) function.

$\succeq_e$	A general grounded ranking, that is a ranking over X, obtained from a social ranking solution (a grounding) *srs* (in particular a ranking obtained through lex-cel).
μ	The function that builds the profile vector of lex-cel.
$\Sigma/\Sigma'/\Sigma_i$	An equivalence class of $\mathcal{P}(X)/\sim$.
c_i^x	The i-th element of $\mu(x)$. Thus, $\mu(x) = (c_1^x, \dots c_k^x)$.
$\succeq_L$	The lexicographic order of vectors.
ale	The anti-lex-cel function.
$\Xi/\Xi'/\Xi_i$	An element equivalence class of $X/\sim_e$.
η	The vector used by anti-lex-cel to build the ranking over $\mathcal{P}(X)$.
c_i^S	The i-th element of $\eta(S)$. Thus, $\eta(S) = (c_1^S, \dots c_q^S)$.
dom	A function that transforms a ranking over the features in F to a ranking over the sets in $\mathcal{P}(X)$. This function is the composition of lex-cel and anti-lex-cel $dom = ale \circ le$.
$\mathfrak{p}$	The preference function that assigns a natural number to each set in $\mathcal{P}(X)$, with regards to the ranking obtained trough *dom*.
S_{pref}	The feasible set in $\mathcal{P}(X)$ that is most preferred with regards to the set ranking $\succeq$ obtained through *dom*.
S_{max}	The feasible set of maximum preference $\mathfrak{p}$.
d_i	The decision variable representing element $x_i \in X$.

Value-aligned norm selection

A	A set of actions.
Ag	A set of agents.
$\mathcal{L}$	A first order language.
φ	A precondition.
θ	A deontic operator (prh/per/obl).
a	An action $a \in A$.
$n/n'/n_i$	A norm.
N	A set of norms.
$v/v'/v_i$	A moral value.
V	A set of moral values.
$\succeq_v$	A ranking over moral values in V.
R	The set of norms relations, $R = \{R_x, R_g\}$.
R_x	The set of exclusive relations, $(n, n') \in R_x$ if n and n' are mutually exclusive norms.
R_g	The set of generalisation relations, $(n, n') \in R_x$ if n generalises n'.
$A(n)$	The ancestors of n, the norms that generalise n.
$S(n)$	The successors of n, the norms that are generalised by n.
$\bar{S}(n)$	The direct successors of n, the norms n' such that $(n, n') \in R_g$.
d_i	The decision variable representing norm $n_i \in N$.

References

1. Allouche, T., Escoffier, B., Moretti, S., Öztürk, M.: Social ranking manipulability for the cp-majority, Banzhaf and lexicographic excellence solutions. In: 29th international joint conference on artificial intelligence and seventeenth pacific rim international conference on artificial intelligence {IJCAI-PRICAI-20}, pp. 17–23 (2020). 10.24963/ijcai.2020/3.
2. Arlegi, R. (2003). A note on bossert, pattanaik and xu's "choice under complete uncertainty: axiomatic characterization of some decision rules." *Economic Theory, 22*(1), 219–225. https://doi.org/10.1007/s00199-002-0270-2.
3. Barberà, S., Bossert, W., Pattanaik, P.K.: Ranking sets of objects. In: Handbook of utility theory, pp. 893–977. Springer (2004). 10.1007/978-1-4020-7964-1_4.
4. Bench-Capon, T.: Value-based reasoning and norms. Artificial Intelligence for Justice pp. 9–17 (2016). https://cgi.csc.liv.ac.uk/~tbc/publications/AI4J_paper_23.pdf
5. Bench-Capon, T.J.M., Atkinson, K.: Abstract argumentation and values. In: Argumentation in Artificial Intelligence, pp. 45–64. Springer (2009). https://doi.org/10.1007/978-0-387-98197-0_3
6. Bernardi, G., Lucchetti, R., & Moretti, S. (2019). Ranking objects from a preference relation over their subsets. *Social Choice and Welfare, 52*(4), 589–606. https://doi.org/10.1007/s00355-018-1161-1.
7. Bossert, W., Pattanaik, P. K., & Xu, Y. (1994). Ranking opportunity sets: An axiomatic approach. *Journal of Economic theory, 63*(2), 326–345. https://doi.org/10.1006/jeth.1994.1045.
8. Bouyssou, D. (1986). Some remarks on the notion of compensation in mcdm. *European Journal of Operational Research, 26*(1), 150–160. https://doi.org/10.1016/0377-2217(86)90167-0.
9. Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly, 69*(1), 9–15. https://doi.org/10.2307/2312726.
10. Greco, S., Ishizaka, A., Tasiou, M., & Torrisi, G. (2019). On the methodological framework of composite indices: A review of the issues of weighting, aggregation, and robustness. *Social Indicators Research, 141*(1), 61–94. https://doi.org/10.1007/s11205-017-1832-9.
11. Haret, A., Khani, H., Moretti, S., Öztürk, M.: Ceteris paribus majority for social ranking. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18, pp. 303–309. international joint conferences on artificial intelligence organization (2018). 10.24963/ijcai.2018/42.
12. Khani, H., Moretti, S., Öztürk, M.: An ordinal Banzhaf index for social ranking. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI-19, pp. 378–384. International joint conferences on artificial intelligence organization (2019). 10.24963/ijcai.2019/54.
13. Klamler, C., Pferschy, U., Ruzika, S.: Committee selection with a weight constraint based on lexicographic rankings of individuals. In: F. Rossi, A. Tsoukias (eds.) Algorithmic Decision Theory, pp. 50–61. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). 10.1007/978-3-642-04428-1_5.
14. Luo, J., Meyer, J.J., Knobbout, M.: Reasoning about opportunistic propensity in multi-agent systems. In: AAMAS 2017 Workshops, Best Papers., pp. 1–16 (2017). 10.1007/978-3-319-71682-4_13.
15. Moretti, S., Öztürk, M.: Some axiomatic and algorithmic perspectives on the social ranking problem. In: International conference on algorithmic decision theory, pp. 166–181. Springer (2017). 10.1007/978-3-319-67504-6_12.
16. Pattanaik, P. K., & Peleg, B. (1984). An axiomatic characterization of the lexicographic maximin extension of an ordering over a set to the power set. *Social Choice and Welfare, 1*(2), 113–122. https://doi.org/10.1007/BF00452883.
17. Roth, A. E., & Sotomayor, M. (1992). Two-sided matching. *Handbook of game theory with economic applications, 1,* 485–541. https://doi.org/10.1016/S1574-0005(05)80019-0.

18. Roy, B., & Słowiński, R. (2013). Questions guiding the choice of a multicriteria decision aiding method. *EURO Journal on Decision Processes, 1*(1), 69–97. https://doi.org/10.1007/s40070-013-0004-7.

19. Santhanam, G. R. (2016). Qualitative optimization in software engineering: A short survey. *Journal of Systems and Software, 111,* 149–156. https://doi.org/10.1016/j.jss.2015.09.001.

20. Serramia, M., Lopez-Sanchez, M., Rodriguez-Aguilar, J.A.: A qualitative approach to composing value-aligned norm systems. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pp. 1233–1241 (2020). http://www.ifaamas.org/Proceedings/aamas2020/pdfs/p1233.pdf

21. Serramia, M., López-Sánchez, M., Rodríguez-Aguilar, J.A., Morales, J., Wooldridge, M., Ansotegui, C.: Exploiting moral values to choose the right norms. In: Proceedings of the 1st conference on artificial intelligence, ethics and society (AIES'18), pp. 1–7 (2018). 10.1145/3278721.3278735.

22. Serramia, M., Lopez-Sanchez, M., Rodriguez-Aguilar, J.A., Rodriguez, M., Wooldridge, M., Morales, J., Ansotegui, C.: Moral values in norm decision making. In: Proceedings of the 17th international conference on autonomous agents and multiagent systems (AAMAS'18), pp. 1294–1302. International Foundation for Autonomous Agents and Multiagent Systems (2018). http://ifaamas.org/Proceedings/aamas2018/pdfs/p1294.pdf

23. Słowiński, R., Greco, S., & Matarazzo, B. (2002). Axiomatization of utility, outranking and decision rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle. *Control and Cybernetics, 31*(4), 1005–1035.