Constraint-Wish and Satisfied-Dissatisfied: An Overview of Two Approaches for Dealing with Bipolar Querying

Tom Matthé[1], Joachim Nielandt[1], Sławomir Zadrożny[2], and Guy De Tré[1]

Abstract In recent years, there has been an increasing interest in dealing with user preferences in flexible database querying, expressing both positive and negative information in a heterogeneous way. This is what is usually referred to as bipolar database querying. Different frameworks have been introduced to deal with such bipolarity. In this chapter, an overview of two approaches is given. The first approach is based on mandatory and desired requirements. Hereby the complement of a mandatory requirement can be considered as a specification of what is not desired at all. So, mandatory requirements indirectly contribute to negative information (expressing what the user does not want to retrieve), whereas desired requirements can be seen as positive information (expressing what the user prefers to retrieve). The second approach is directly based on positive requirements (expressing what the user wants to retrieve), and negative requirements (expressing what the user does not want to retrieve). Both approaches use pairs of satisfaction degrees as the underlying framework but have different semantics, and thus also different operators for criteria evaluation, ranking, aggregation, etc.

Key words: constraint-wish approach, satisfied-dissatisfied approach, bipolar database querying

[1]Department of Telecommunications and Information Processing,
Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
e-mail: {Guy.DeTre,Tom.Matthe,Joachim.Nielandt}@UGent.be
[2]Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw, Poland
e-mail: Slawomir.Zadrozny@ibspan.waw.pl

1 Introduction

In daily life, it can be observed that people, whilst communicating their preferences, tend to use vague or fuzzy terms in expressing their desires. A typical example is a recruitment office that, e.g., is searching for *young* people with a *high* score in math. A lot of research has been done to translate this 'fuzziness' to the domain of database querying, resulting in 'fuzzy' querying of regular databases, where the queries are composed of several 'fuzzy' query conditions, interconnected by logical connectives. Indeed, the main lines of research in this area include the study of modeling linguistic terms (like, e.g., *young* or *high*) in the specification of elementary query conditions using elements of fuzzy logic [38] and the enhancement of fuzzy query formalism with soft aggregation operators [23, 22, 6, 15]. Both linguistic terms and soft aggregations model user's preferences [4] and, as such, require a query satisfaction modeling framework that supports rank-ordering the records retrieved in response to a query according to the degree to which they satisfy all conditions imposed by the query. Usually, query satisfaction in 'fuzzy' querying of regular databases is modelled by associating a *satisfaction degree s* with each record in the answer set of the query. These satisfaction degrees take values in the unit interval $[0,1]$ and are computed during query processing. The value 0 means complete lack of satisfaction and implies that the associated record does not belong to the query's answer set. The value 1 expresses full satisfaction, while all other, intermediate, values denote partial query satisfaction. Records with a satisfaction degree s that is lower than a given threshold value δ, i.e., for which $s < \delta$, are usually discarded from the query answer set.

A more advanced aspect of specifying user preferences in database queries concerns the handling of *bipolarity*. Bipolarity hereby refers to the fact that users might distinguish between positive and negative aspects (or between constraints and wishes) while specifying their query preferences. Positive statements may be used to express what is possible, satisfactory, permitted, desired or acceptable, whereas negative statements may express what is impossible, unsatisfactory, not permitted, rejected, undesired or unacceptable. Likewise, constraints express what is accepted, whereas wishes are used to specify which of the accepted values are really desired by the user. Bipolarity is inherent to human communication and natural language and should hence be reflected and dealt with in any querying system that aims to support human interaction as adequate as possible.

For example, consider the specification of user preferences in the context of selecting a car, more specifically concerning the color of a car. A positive statement is 'I like black or dark blue cars', while 'I do not want a white car' is a negative statement. In terms of constraints and desires, similar preferences might be expressed by 'I want a dark colored car' and 'if possible, I really prefer a black or dark blue car'. Remark that often, negative conditions might be translated to constraints, while positive conditions might be seen as wishes.

Depending on the situation, it may be more natural for a user to use negative conditions or positive conditions. Sometimes one can use both positive and negative conditions at the same time. This is especially the case if the user does not have

complete knowledge of the domain on which the criterion is specified, or if this domain is too large to completely specify the user's preferences for every value in the domain, as can for example be the case with available car colors.

In standard approaches to regular 'fuzzy' querying it is explicitly assumed that a record that satisfies a query condition to a degree s, at the same time dissatisfies it, i.e., satisfies its negation, to a degree $1-s$. This assumption does not generally hold when dealing with bipolar query criteria specifications as positive and negative conditions comprising a query are assumed to be independent, i.e., may assume any value from the interval [0,1]. In such situations of *heterogeneous bipolarity*, a semantically richer query satisfaction modeling approach, which is more consistent with human reasoning and is able to model this bipolarity, is preferred.

In this chapter, two such approaches to bipolar database querying are discussed. On the one hand, the *constraint-wish* (or mandatory-desired) approach will be presented, used amongst others by Dubois and Prade [16, 17] and Bosc et al. [10, 26, 27, 29, 39], and on the other hand, the *satisfied-dissatisfied* (or positive-negative) approach will be discussed, used amongst others by Zadrożny et al. [47] and De Tré et al. [13, 30, 33]. For both approaches, an overview is given, which consecutively handles the semantics of the actual framework, the evaluation of query conditions within this framework, the ranking of query results and the aggregation of compound query conditions.

The remainder of this chapter has been organised as follows: first, some preliminaries on bipolar query conditions will be presented in Section 2, explaining the two approaches that will be discussed in this chapter in more detail, together with their semantics. The next Section 3 discusses the ranking of the results of a bipolar query. Next, in Section 4, different techniques to aggregate the results of multiple query conditions are presented. Finally, Section 5 states some conclusions.

2 Bipolar Query Conditions

Pioneering work in the area of heterogeneous bipolar database querying has been done in [25], which seems to be the first approach where a distinction has been made between mandatory query conditions and desired query conditions. As mentioned earlier, desired and mandatory conditions can be viewed as specifying positive and negative information, respectively. Indeed, the opposite of a mandatory condition specifies what must be rejected and thus what is considered as being negative with respect to the query result, whereas desired conditions specify what is considered as being positive.

Later on, this idea has been further developed and adapted to be used in 'fuzzy' querying techniques. The use of the twofold fuzzy sets (TFS) to represent a bipolar elementary query condition with respect to a given attribute A is reported, e.g., in [16, 17]. A twofold fuzzy set expresses which domain values are accepted by the user and which among these accepted values are really desired by her or him. An alternative approach, based on the concept of an Atanassov (intuitionistic) fuzzy set

(AFS) and departing from the specification of which values are desired and which values are undesired, is presented in [12, 30]. Both approaches have in common that they deal with bipolarity that is specified *inside* elementary query conditions, i.e., in the domain of an attribute three subsets, in general fuzzy, are distinguished: of positively, negatively and neutrally evaluated elements.

Other approaches study bipolarity that is specified *between* elementary query conditions, meaning that these conditions are assigned different semantics. In particular, a distinction can be made between mandatory and desired query conditions. These conditions can still contain vague terms modelled by fuzzy sets as in regular 'fuzzy' querying [38, 22, 5, 7, 21, 4, 46]. For example, in [48] an approach is presented where bipolar queries are represented as a special case of the fuzzy 'winnow' operator. Bipolarity is thus studied considering queries with preferences as in [25]. An alternative assumption that can be made, is considering queries that consist of a number of 'positive' and 'negative' elementary conditions [13, 33].

In this chapter, only bipolarity that is specified *inside* elementary query conditions will be considered. In the following subsections, two approaches will be discussed in more detail: the *constraint-wish* and the *satisfied-dissatisfied* approach.

2.1 Constraint-Wish Approach

Consider a universe of discourse U corresponding to the domain of an attribute in question. In the *constraint-wish* approach (referred to elsewhere in this volume also as 'required-desired semantics'), the bipolar query condition consists of two parts: a constraint C, which describes the set of acceptable values of U, and a wish W, which defines the set of wished-for (or desired) values of U. In general, the constraint and the wish are specified using fuzzy sets C and W, defined on U [44], identified by their respective membership functions μ_C and μ_W. Because it is not coherent to wish something that is rejected (where the rejected values are represented by the complement of the fuzzy set C), a consistency condition is imposed. Two forms of consistency conditions may be considered:

- *Strong consistency*,

$$\forall x \in U : \mu_C(x) < 1 \Rightarrow \mu_W(x) = 0. \tag{1}$$

  In this case, the support of the wish W is required to be a subset of the core of the constraint C, which means that the wish can play any role in evaluating only those records which fully satisfy the constraint. The pair of fuzzy sets C and W then form a *twofold fuzzy set* [11].
- *Weak consistency*,

$$\forall x \in U : \mu_W(x) \leq \mu_C(x). \tag{2}$$

  In this case, the wish is required to be more specific than the constraint what represents the fact that it is harder to satisfy a wish than to satisfy a constraint, but

the wish can also play a role in evaluating the records which do not fully satisfy the constraint. The pair of fuzzy sets C and W then form an interval-valued fuzzy set (IVFS) [20, 36, 45, 19].

Because a twofold fuzzy set is formally a special case of an interval-valued fuzzy set, the bipolar query condition can in both cases be modelled by means of an IVFS, which is defined by

$$F = \{(x, [\mu_{F_*}(x), \mu_{F^*}(x)]) | (x \in U) \wedge (0 \leq \mu_{F_*}(x) \leq \mu_{F^*}(x) \leq 1)\}. \tag{3}$$

Thus, a bipolar query condition is modelled by means of an IVFS, where the upper membership function μ_{F^*} models the constraint, i.e., $\mu_{F^*} = \mu_C$, and the lower membership function μ_{F_*} models the wish, i.e., $\mu_{F_*} = \mu_W$.

An important feature of this semantics is that the wish plays somehow a secondary role in the query. A bipolar query condition in the constraint-wish approach should be interpreted as '*satisfy C and, if possible, satisfy W*' [16].

Summarising, in this approach the evaluation of a record R against an elementary bipolar query condition 'A IS F' with F composed of a couple (C,W) of fuzzy sets C and W results in a pair of satisfaction degrees $(c(R), w(R)) \in [0,1]^2$ such that

$$c(R) = \mu_C(R[A]) \tag{4}$$

$$w(R) = \mu_W(R[A]) \tag{5}$$

where $R[A]$ denotes the value of record R for attribute A.

2.2 Satisfied-Dissatisfied Approach

In the *satisfied-dissatisfied* approach, the bipolar query condition also consists of two parts. One part specifies the values of an attribute A which are positively evaluated by the user with respect to her or his preferences and, *independently*, another part specifies the values for A which are negatively evaluated by the user. A pair of fuzzy sets, F^+ and F^-, expressing the respective parts of the query condition may be treated as a bipolar extension to the concept of fuzzy set. *Atanassov (intuitionistic) Fuzzy Sets* (AFSs) [1] are an example of such an extension. An AFS F over a universe U is formally defined by

$$F = \{(x, \mu_F(x), \nu_F(x)) | (x \in U) \wedge (0 \leq \mu_F(x) + \nu_F(x) \leq 1)\}. \tag{6}$$

where $\mu_F : U \to [0,1]$ and $\nu_F : U \to [0,1]$ are respectively called the membership and non-membership degree functions and $0 \leq \mu_F(x) + \nu_F(x) \leq 1$, $\forall x \in U$ reflects the consistency condition of the AFS. In the context of database querying, this consistency condition can be interpreted as stating that the degree of non-preference $\nu_F(x)$ for a given value x can never be larger than the complement $1 - \mu_F(x)$ of the degree of preference for that value (or, equivalently, that the degree of preference

$\mu_F(x)$ for a value x can never be larger than the complement $1-\nu_F(x)$ of the degree of non-preference for that value.)

Formally, in their basic form, AFSs are operationally equivalent to IVFSs and thus may be also used to represent preferences in the constraint-wish approach, but their intended semantics is closer to the idea of the satisfied-dissatisfied approach. However, in the satisfied-dissatisfied approach the total independence of positive and negative condition is assumed and the AFS's consistency condition does not meet this assumption. Thus, in what follows we will use the concept *bipolar AFS* which follows the two membership functions structure of AFSs but drops the consistency condition. In this respect, the presented approach is similar to the neutrosophic logic [37, 35]. It is also similar to a fuzzy version of Belnap's logic [2], proposed by Öztürk and Tsoukias [34] and further developed by Turunen et al. [40]; cf. also a study on links between Belnap's logic and bipolarity by Konieczny et al. [24]. However, it should be stressed that the degrees of satisfaction and dissatisfaction do not have any epistemic flavour here, i.e., e.g., they do not form an interval containing a 'true' degree to which the user likes the given value of an attribute in question. Instead, these degrees respectively express the genuine liking and disliking of the value which are assumed to occur simultaneously and independently of each other.

In what follows, we will thus often adopt the notation μ_F and ν_F instead of, respectively μ_{F^+} and μ_{F^-}, while referring to the sets of positively and negatively evaluated values of an attribute under consideration.

We have thus a couple $(\mu_F(x), \nu_F(x))$ which is referred to as the *bipolar satisfaction degree* (BSD) and represents the suitability of $x \in dom_A$ with respect to a condition A IS F, where A is an attribute and F is a bipolar AFS representing preferences of the user. Now, the question is how these couples are to be processed, i.e., used to order the records in an answer to the query and aggregated with the couples related to other elementary conditions. We discuss these issues in the following sections. An elementary bipolar query condition 'A IS F' in the satisfied-dissatisfied approach should be interpreted as '*preferably satisfy* F^+ *and preferably do not satisfy* F^-' [30].

Summarising, in this approach the evaluation of a record R against an elementary bipolar query condition 'A IS F' with F a bipolar AFS characterized by a pair (μ, ν) of membership functions μ and ν results in a pair $(s(R), d(R)) \in [0,1]^2$ of values, referred to as *satisfaction degree* (s) and *dissatisfaction degree* (d), jointly called a Bipolar Satisfaction Degree (BSD) [30], such that

$$s(R) = \mu_F(R[A]) \tag{7}$$

$$d(R) = \nu_F(R[A]) \tag{8}$$

where $R[A]$ is the value of record R for attribute A. The set of all possible BSDs will be denoted as $\tilde{\mathbb{B}}$.

2.3 Examples

As an example of an elementary bipolar query condition, consider the case of a real estate application and a user who wants to find a suitable house to buy. An important criterion may be the size of the garden. The user may have a number of criteria in mind when judging which ranges of values of this attribute she or he prefers. For example, considering garden as a playground for children the user may use a positive unipolar scale to measure its suitability – the larger the size the better. On the other hand, taking into account the maintenance costs of the garden the user may use a negative unipolar scale – the larger the garden size the higher the costs.[1]. Let us assume that the terms 'large' and 'too large', respectively, represent the preferences of the user along these two criteria and thus describe the sets F^+ and F^- of positive and negative parts of the bipolar condition.

Figure 1(a) shows how such preferences may be represented in the framework of the satisfied-dissatisfied approach. It is worth noticing that, for example, a garden size of 550 sq. m. is totally negatively evaluated from the point of view of the maintenance costs and, at the same time, totally positively evaluated from the point of view of fun for the children.

Looking for a counterpart in the constraint-wish approach we would like to interpret the positive condition as a wish and the negative condition as the complement of the constraint. However, this is not possible as the consistency condition implied by the semantics of the constraint-wish approach is not met, what is illustrated in Figure 1(b), i.e., there are some values x where, $\mu_W(x) > \mu_C(x)$.

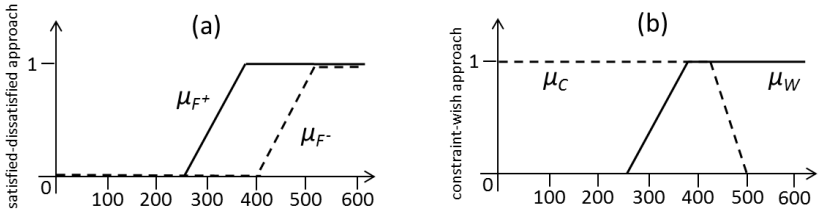

Fig. 1 Examples: (a) μ_{F^-}: 'too large', μ_{F^+}: 'large'; (b) μ_C: 'not too large', μ_W: 'large'.

In order to illustrate the constraint-wish approach at work let us assume the following scenario. The user may look for a 'large' garden but she or he would be most happy with a garden of size around 400-500 sq. m. Thus, the former may be interpreted as a constraint ('not large' garden is excluded) while the latter is just a desired size. Figure 2(b) shows an example of membership functions which may serve to represent such preferences in the framework of the constraint-wish approach.

[1] We are slightly simplifying the situation here as with respect to both criteria the user may have in mind two separate bipolar scales, but still it will result in sets of aggregated positively and negatively evaluated garden size values.

It should be stressed that the satisfied-dissatisfied approach is not suitable to represent such preferences. One can consider a kind of representation shown in Figure 2(a) which is obtained by treating the wish and the constraint as, respectively, the positive evaluation and the complement of the negative evaluation. However, in the framework of the satisfied-dissatisfied approach Figure 2(a) should be actually interpreted as representing the following preferences: the user has positive feelings about the garden size being ca. 400-500 sq. m. and does not like small gardens (more precisely: not large gardens).

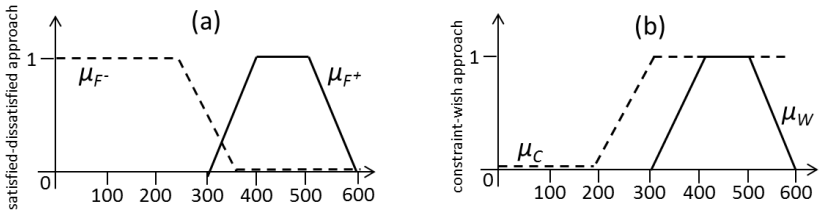

Fig. 2 Examples: (a) μ_{F^-}: 'not large', μ_{F^+}: 'around 400-500 sq. m.'; (b) μ_C: 'large', μ_W: 'around 400-500 sq. m'.

Figure 3(b) shows a slightly different wish which may be expressed as 'preferred size of the garden is ca. 400-500 sq. m. or slightly less'. This case is still well suited to be represented in the constraint-wish based approach although only the weak consistency is preserved.

A kind of the counterpart of the above in the framework of the satisfied-dissatisfied approach, again in the spirit of Figure 2(a), is shown in Figure 3(a).

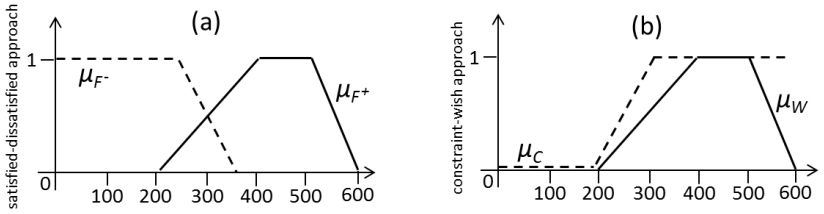

Fig. 3 Examples: (a) μ_{F^-}: 'not large', μ_{F^+}: 'around 400-500 sq. m. or slightly less'; (b) μ_C: 'large', μ_W: 'around 400-500 sq. m. or slightly less'.

3 Ranking of Query Results

After evaluating a bipolar query condition for all potential query results, every resulting record R_i will have an associated pair of calculated satisfaction degrees, either a pair $(c(R_i), w(R_i))$ or a BSD $(s(R_i), d(R_i))$. Now, we will deal with the question how the records should be ranked in the response to a query using these pairs of degrees.

3.1 Ranking in the Constraint-Wish Approach

In this approach it is assumed that constraints and wishes are not compensatory [10, 26], i.e., a higher satisfaction of a wish can not compensate a lower satisfaction of a constraint. Therefore, ranking is done primarily on the constraint satisfaction, and secondly, in case of ties, on the wish satisfaction. In general, one has [18]:

$$R_1 \succ R_2 \Leftrightarrow (c(R_1) > c(R_2)) \vee (c(R_1) = c(R_2) \wedge w(R_1) > w(R_2)) \tag{9}$$

where $R_1 \succ R_2$ means that R_1 is preferred to R_2. Thus, this is the lexicographical ordering with respect to the pairs (c, w).

Another possibility is scalarization: a real function may be applied to the pairs $(c(R_i), w(R_i))$ and the records are then ranked according to the values obtained. Zadrożny and Kacprzyk [48], following Lacroix and Lavency [25], propose the aggregation of both degrees in the spirit of the 'and possibly' operator. In this approach the wish is taken into account only 'if possible', i.e., if its satisfaction does not interfere with the satisfaction of the constraint what is determined with respect to the content of the whole database. The same idea, applied in a different context, may be found in some earlier work of Bordogna and Pasi [3], Dubois and Prade [14] or Yager [41]. Recently, a lot of work has been done on the study of different interpretations of the 'and possibly' as well as its dual 'or at least' operators by Bosc, Pivert, Tamani, Hadjali (see, e.g., [8, 28, 9]) and by Dubois and Prade in this volume.

3.2 Ranking in the Satisfied-Dissatisfied Approach

Because, in the satisfied-dissatisfied approach, the satisfaction degree and the dissatisfaction degree are assumed to be totally independent, both should have an equal impact on the ranking [30]. Naturally, the higher the satisfaction degree, the higher the ranking should be, and dually, the higher the dissatisfaction degree, the lower the ranking should be. A possible ranking function r for BSDs (s, d), with a complete symmetrical impact of both the satisfaction and dissatisfaction degrees, is the following:

$$r(s,d) = \frac{s+(1-d)}{2}. \tag{10}$$

This ranking function produces values in $[0,1]$. Three special cases can be distinguished:

- $r(s,d) = 1$: in this case it must be that $s = 1$ and $d = 0$, so this is the case of *full global satisfaction*.
- $r(s,d) = 0$: in this case it must be that $s = 0$ and $d = 1$, so this is the case of *full global dissatisfaction*.
- $r(s,d) = 0.5$: in this case it must be that $s = d$ and the ranking can be considered *neutral*. The condition is as satisfied as it is dissatisfied.

Remark that both degrees equally matter when ranking the records, as expected. For example, records for which the evaluation leads to a dissatisfaction degree $d = 1$, or dually a satisfaction degree of $s = 0$, should not a priori be excluded as being totally unsatisfactory. Indeed, e.g., the BSDs $(1,1)$ and $(0,0)$, although having $d = 1$ (respectively $s = 0$), both have neutral ranking ($r(s,d) = 0.5$) and are hence situated in the middle of the ranking spectrum.

Other ranking functions are also possible, e.g., assigning more importance to either the satisfaction degree or the dissatisfaction degree. In general, a suitable ranking function r for BSDs should meet the following minimal requirements:

1. $0 \leq r(x,y) \leq 1$, with (x,y) a BSD, i.e., $r : \tilde{\mathbb{B}} \to [0,1]$.
2. $r(1,0) = 1$, i.e., the BSD with full satisfaction and no dissatisfaction should be ranked the highest.
3. $r(0,1) = 0$, i.e., the BSD with full dissatisfaction and no satisfaction should be ranked the lowest.
4. $\forall x,y \in [0,1] : r(x,x) = r(y,y)$, i.e., for all BSDs with equal satisfaction degree and dissatisfaction degree, the ranking should also be equal. The reason for this requirement is that, ranking wise, it is impossible to make a sensible distinction between the cases of total indifference (i.e., BSD $(0,0)$) and total conflict (i.e., BSD $(1,1)$), and also all other intermediate cases where $s = d$ (i.e., BSD (x,x), $x \in [0,1]$).
5. monotonicity: $r(x,y) \leq r(x+\varepsilon,y)$ and $r(x,y) \geq r(x,y+\varepsilon)$.

These minimal requirements eliminate the use of ranking functions which solely rank on either the satisfaction degree s or the dissatisfaction degree d, and use the other degree (d or s respectively) only as a 'tiebreaker', because they would violate the fourth requirement. Thus, for example, the lexicographical ordering, meaningfully used in the constraint-wish approach (see Eq. (9)), is not suitable for the satisfied-dissatisfied approach because of the assumed total independence between, and equally important role of, the satisfaction and dissatisfaction degrees.

A list of useful ranking functions for BSDs is listed below:

$$r_1 = \frac{s+(1-d)}{2} \tag{11}$$

$$r_2 = \frac{s}{s+d} \tag{12}$$

$$r_3 = \frac{1-d}{(1-s)+(1-d)} \tag{13}$$

$$r_4 = \frac{s}{s+d} \cdot \frac{1-d}{(1-s)+(1-d)} \tag{14}$$

$$r_5 = \max\{0, s-d\} \tag{15}$$

$$r_6 = \min\{1+s-d, 1\}. \tag{16}$$

Ranking function r_2 is discontinuous in BSD $(0,0)$, r_3 is undefined for BSD $(1,1)$, while r_4 is undefined for BSDs $(0,0)$ and $(1,1)$. More information on the behaviour and properties of these ranking functions can be found in [32].

3.3 Comparison and discussion

The lexicographical ordering used in the constraint-wish approach makes wishes (positive information) rather secondary in comparison to the constraints (negative information), according to the assumed semantics. This may be, however, counter-intuitive in some cases. Let us consider two pairs of degrees $(c(R_1), w(R_1))$ and $(c(R_2), w(R_2))$ such that $c(R_1) = c(R_2) + \varepsilon$, while $w(R_1) = 0$ and $w(R_2) = 1$. In such a case R_1 will be ranked before R_2, even for ε very close to 0 what may be disputable. A possible escape is to assume a discrete scale for c's and w's with a small number of levels and to claim that the smallest difference in levels of the constraint satisfaction is large enough to justify its definite role in establishing the ranking of records whatever their satisfaction of wishes is.

A scalarization in the spirit of the 'and possibly' operator is an interesting option but it adopts a specific semantics of constraints and wishes.

Ranking in the satisfied-dissatisfied approach is based on the ranking of the BSDs. Due to their specific semantics and the total independence of the satisfaction and dissatisfaction degrees, BSDs can be ranked in different ways. A ranking function for BSDs should satisfy the requirements specified in Subsection 3.2. The selection of a ranking function depends on the requirements of the application.

- If it is necessary to assign an equal weight to $s(R)$ and $d(R)$, then the ranking function r_1 (cf. Eq. (11)) can be used. In this approach, query conditions are interpreted as preferences because records with $d(R) = 1$ or $s(R) = 0$ are not a priori excluded from the result, i.e., they do not necessarily result in a ranking value 0.
- If the non-zero satisfaction degree should be interpreted as an absolute requirement, i.e., if $s(R) = 0$ has to imply that the ranking value is 0, then the ranking function r_2 (cf. Eq. (12)) can be used.

- Dually, if avoiding the total dissatisfaction should be interpreted as an absolute requirement, i.e., if $d(R) = 1$ has to imply a ranking value 0, then the ranking function r_3 (cf. Eq. (13)) can be used.
- Ranking function r_4 (cf. Eq. (14)) can be used if both non-zero satisfaction degree is required and total dissatisfaction should be avoided.
- Finally, if the ranking should be based on the best of $s(R)$ and $d(R)$, then either ranking function r_5 (cf. Eq. (15)) or ranking function r_6 (cf. Eq. (16)) can be used. Hereby, $r_5 = 0$ if $s(R) \leq d(R)$ and $r_6 = 1$ if $s(R) \geq d(R)$.

It is worth noting that modelling bipolarity *inside* an elementary query condition using the constraint-wish approach (cf. Section 2.1) makes the ranking problem somehow trivial. Namely, it is easy to verify that due to the consistency condition, it is impossible to have two pairs of degrees $(c(R_1), w(R_1))$ and $(c(R_2), w(R_2))$ such that $c(R_1) < c(R_2)$ and at the same time $w(R_1) > w(R_2)$. This further justifies the primary role of the constraint satisfaction degree in the ranking process, as defined in (9). On the other hand, this is not the case in the satisfied-dissatisfied approach what makes room for more possible definitions of ranking.

4 Aggregation in Bipolar Query Processing

So far we have focused on bipolar queries comprising one elementary bipolar condition with respect to an attribute. In what follows we consider a compound bipolar query composed of many elementary bipolar conditions, possibly combined using explicit logical connectives of conjunction, disjunction and negation. The evaluation of an elementary bipolar query condition A IS F results in a pair of degrees (either $(c(R), w(R))$ or $(s(R), d(R))$) for every database record R. Now, consider the evaluation of an entire query composed of n elementary query conditions. First, for each relevant database record R, each elementary condition need to be evaluated resulting in n individual pairs of satisfaction degrees. Second, all these individual pairs must be *aggregated* to come up with a global result reflecting the extent to which R satisfies the entire bipolar query. The basic aggregation techniques in case of the constraint-wish approach and the satisfied-dissatisfied approach are presented in the two subsections below. A distinction has been made between techniques where the pairs of degrees are treated as a whole and techniques where these degrees are treated individually.

4.1 Aggregation in the Constraint-Wish Approach

Consider n elementary bipolar query conditions, the evaluation of which for a record R leads to a set of n pairs $(c_i(R), w_i(R)), i = 1, \ldots, n$. This set of n pairs needs to be aggregated to obtain the global satisfaction degree.

4.1.1 Treating $c(R)$ and $w(R)$ Individually

In this approach a bipolar query is meant as a list of elementary bipolar conditions and their conjunction is tacitly assumed. The $c_i(R)$'s and $w_i(R)$'s are separately aggregated [16, 17]. Both aggregations are guided appropriately by the semantics of the constraints and of the wishes. Namely, it is assumed that, if a record R does not satisfy a constraint then it should be rejected overall. Therefore, the degrees $c_i(R)$ are aggregated in a conjunctive way. On the other hand, if a record is desirable according to one wish then it is desirable overall. Therefore, the degrees $w_i(R)$ are aggregated in a disjunctive way. This then leads to a global pair $(c(R), w(R))$ expressing the satisfaction of the whole bipolar query by a record R:

$$(c(R), w(R)) = (\min_i c_i(R), \max_i w_i(R)). \tag{17}$$

Besides the minimum and maximum, other aggregation operators, based on triangular norms and co-norms, can also be used if a reinforcement effect is needed or desired.

Remark that, in general, this aggregation technique will not preserve consistency, i.e., it is possible that $w(R) > c(R)$. This can be solved by treating the 'global wish' not just as the mere disjunction of all wishes, but by also taking the conjunction of this disjunction with all the constraints [17]:

$$(c(R), w(R)) = (\min_i c_i(R), \min(\max_i w_i(R), \min_i c_i(R))). \tag{18}$$

4.1.2 Treating $(c(R), w(R))$ as a Whole

This approach, followed amongst others by Bosc et al. [26, 10], does not look at the $c_i(R)$'s and $w_i(R)$'s separately, but treats them as a whole. In contrast with Dubois and Prade, Bosc et al. consider both conjunction and disjunction of bipolar query conditions. As it is usually done in regular 'fuzzy' querying, conjunction is translated to a minimum operator and disjunction is translated to a maximum operator. In order to take the minimum or maximum, the set of $(c_i(R), w_i(R))$ pairs must be ordered. In this approach, a lexicographical ordering is assumed (see above, in Subsection 3.1) and the operators *lmin* and *lmax* are introduced as aggregation operators for respectively conjunction and disjunction of bipolar query conditions [17, 26, 10]. Let us assume that two elementary bipolar queries A_i IS F_i, $i = 1, 2$, result in two pairs of satisfaction degrees for a record R: $(c_i(R), w_i(R))$, $i = 1, 2$. Then, the pair of satisfaction degrees for the conjunction and disjunction of these elementary queries is defined as follows:

$$
\begin{aligned}
&(c_{(A_1 \mathrm{IS} F_1)\wedge(A_2 \mathrm{IS} F_2)}(R), w_{(A_1 \mathrm{IS} F_1)\wedge(A_2 \mathrm{IS} F_2)}(R)) = \\
&\qquad = lmin((c_1(R), w_1(R)), (c_2(R), w_2(R)) = \\
&= \begin{cases} (c_1(R), w_1(R)) \text{ if } (c_1(R) < c_2(R)) \vee (c_1(R) = c_2(R) \wedge w_1(R) < w_2(R)) \\ (c_2(R), w_2(R)) \text{ otherwise} \end{cases}
\end{aligned} \tag{19}
$$

$$
\begin{aligned}
&(c_{(A_1 \mathrm{IS} F_1)\vee(A_2 \mathrm{IS} F_2)}(R), w_{(A_1 \mathrm{IS} F_1)\vee(A_2 \mathrm{IS} F_2)}(R)) = \\
&\qquad = lmax((c_1(R), w_1(R)), (c_2(R), w_2(R)) = \\
&= \begin{cases} (c_1(R), w_1(R)) \text{ if } (c_1(R) > c_2(R) \vee (c_1(R) = c_2(R) \wedge w_1(R) > w_2(R)) \\ (c_2(R), w_2(R)) \text{ otherwise.} \end{cases}
\end{aligned} \tag{20}
$$

Due to the associativity of the operators *lmin* and *lmax* formulas (19) and (20) may be easily extended to the case of a conjunction and disjunction, respectively, of n elementary bipolar queries.

By definition both *lmin* and *lmax* return one of the input pairs as the result. As all arguments are assumed to be consistent so is also the result of this type of aggregation.

4.2 Aggregation in the Satisfied-Dissatisfied Approach

Consider again n bipolar query conditions, evaluation of which for record R leads to a set $\{(s_i(R), d_i(R)), i = 1, \ldots, n\}$ of n BSDs. This set of n pairs needs to be aggregated to a BSD $(s(R), d(R))$ representing the global satisfaction and dissatisfaction when taking into account all imposed query conditions.

4.2.1 Treating s and d Individually

In this approach, as in the approach by Dubois and Prade, the BSDs are not aggregated as a whole but the lists of $s_i(R)$'s and $d_i(R)$'s are aggregated separately [30, 33]. But, unlike the Dubois and Prade approach, both conjunction and disjunction of bipolar query conditions are considered, as well as the negation. Moreover, this approach also allows to take into account weights to distinguish important from less important query conditions.

Because the bipolar query conditions in this approach are inspired by AFSs, the basic aggregation of BSDs (which are the result of the evaluation of such bipolar query conditions) is also inspired by the aggregation of AFSs. This means that the conjunction (respectively disjunction) of two BSDs is calculated in the same sense as the intersection (respectively union) of two AFSs. Moreover, these operations also coincide with those proposed in a continuous extension of Belnap's four-valued logic proposed by Öztürk and Tsoukiàs [34].

Non-Weighted Aggregation

Let us consider two elementary bipolar conditions: 'A_1 IS F_1' and 'A_2 IS F_2', and their conjunction and disjunction.

Conjunction.

The satisfaction and dissatisfaction degrees, i.e., a BSD, for the query '$(A_1$ IS $F_1) \wedge (A_2$ IS $F_2)$' is computed as follows:

$$(s_{(A_1 \text{ IS } F_1)\wedge(A_2 \text{ IS } F_2)}(R), d_{(A_1 \text{ IS } F_1)\wedge(A_2 \text{ IS } F_2)}(R)) = \\ = (\min(s_{A_1 \text{ IS } F_1}(R), s_{A_2 \text{ IS } F_2}(R)), \max(d_{A_1 \text{ IS } F_1}(R), d_{A_2 \text{ IS } F_2}(R))). \quad (21)$$

An intuitive justification for this formula is as follows:

- For the conjunction of two conditions to be satisfied, both conditions have to be satisfied. Therefore the minimum of both individual satisfaction degrees is taken as the satisfaction degree of their conjunction.
- For the conjunction to be dissatisfied, it is enough if one of them is dissatisfied. Therefore the maximum of both individual dissatisfaction degrees is taken as the dissatisfaction degree of their conjunction.

Besides the minimum and maximum, other aggregation operators based on triangular norms and co-norms can also be used if a reinforcement effect is needed or desired.

It should be noted that the formulas (17) and (21) although similar on the surface, are quite different. In both cases we have the minimum operator applied to the first components of the aggregated pairs and the maximum operator applied to the second components of these pairs. However, in the former case the minimum and maximum operators are applied to the complements of the negative evaluations and the positive evaluations, respectively, while in the latter case these are positive and negative evaluations, respectively.

Disjunction.

The satisfaction and dissatisfaction degrees, i.e., a BSD, for the query '$(A_1$ IS $F_1) \vee (A_2$ IS $F_2)$' is computed as follows:

$$(s_{(A_1 \text{ IS } F_1)\vee(A_2 \text{ IS } F_2)}(R), d_{(A_1 \text{ IS } F_1)\vee(A_2 \text{ IS } F_2)}(R)) = \\ = (\max(s_{A_1 \text{ IS } F_1}(R), s_{A_2 \text{ IS } F_2}(R)), \min(d_{A_1 \text{ IS } F_1}(R), d_{A_2 \text{ IS } F_2}(R))). \quad (22)$$

Similarly to the case of conjunction, an intuitive justification for this formula is as follows:

- For the disjunction of two conditions to be satisfied, it is enough for one of them to be satisfied. Therefore the maximum of both individual satisfaction degrees is taken as the satisfaction degree of their disjunction.
- For the disjunction of two conditions to be dissatisfied, both of them have to be dissatisfied. Therefore the minimum of both individual dissatisfaction degrees is taken as the dissatisfaction degree of their disjunction.

Negation.

The satisfaction and dissatisfaction degrees, i.e., a BSD, for the query '$\neg$ (A IS F)' is computed as follows:

$$(s_{\neg(A \text{ IS } F)}(R), d_{\neg(A \text{ IS } F)}(R)) = (d_{A \text{ IS } F}(R), s_{A \text{ IS } F}(R)). \tag{23}$$

The same effect of negation can also be achieved by swapping fuzzy sets of positively (F^+) and negatively (F^-) evaluated elements of dom_A composing F, $F = (F^+, F^-)$ in '$\neg$ (A IS F)', i.e., '$\neg$(A IS F)' is thus equivalent to 'A IS F'', where $F' = (F^-, F^+)$.

Weighted Aggregation

When expressing queries (bipolar or not), one way to model the difference in importance between different elementary (bipolar) query conditions is by using weights. Also in the framework of BSDs, it is possible to deal with such weights [31]. The underlying aggregation operators are still appropriate basic aggregation operators, but a premodification step is performed on the elementary criteria evaluation results to take into account the impact of the weights. It is assumed that the importance of a condition, with respect to the final result, is linked with the condition itself, not with the degree to which the condition is satisfied. So weights $w_i \in [0,1]$ can be attached to the individual elementary bipolar conditions. The semantics of the weights is as follows: $w_i = 1$ denotes that the condition is fully important, while $w_i = 0$ denotes that the condition is not important at all. Such a condition can be neglected (and hence should have no impact on the result). Conditions with intermediate weights should still be taken into account, but to a lesser extent than conditions with weight $w_i = 1$. In order to have an appropriate scaling, it is assumed that $\max_i w_i = 1$ [15].

To reflect the impact of a weight on the evaluation of a condition, a premodification is performed on the initial BSDs, taking into account the weights. This means that, before aggregating the individual BSDs, the impact of the weights on these BSDs is calculated first. Afterwards, the modified BSDs are aggregated using the regular aggregation techniques, as if they were regular, non-modified, BSDs. Let g be the operator that models this weight influence on the individual BSDs:

$$g : [0,1] \times \tilde{\mathbb{B}} \to \tilde{\mathbb{B}} : (w, (s,d)) \mapsto g(w, (s,d)). \tag{24}$$

It has been shown that implication functions f_{im} and co-implication functions f_{im}^{co} can be used to model the impact of weights, where f_{im} and f_{im}^{co} are $[0,1]$-valued extensions of Boolean implication and co-implication functions. As an example, consider the Kleene-Dienes implication and co-implication:

$$\begin{aligned} f_{im_{KD}}(x,y) &= \max(1-x,y) \\ f_{im_{KD}}^{co}(x,y) &= \min(1-x,y). \end{aligned} \tag{25}$$

The impact of a weight on a BSD, in case of conjunction, can be defined as follows:

$$g^{\wedge} : [0,1] \times \tilde{\mathbb{B}} \to \tilde{\mathbb{B}} : (w,(s,d)) \mapsto g^{\wedge}(w,(s,d)) = \left(s_{g^{\wedge}(w,(s,d))}, d_{g^{\wedge}(w,(s,d))}\right) \tag{26}$$

where

$$\begin{aligned} s_{g^{\wedge}(w,(s,d))} &= f_{im}(w,s) \\ d_{g^{\wedge}(w,(s,d))} &= f_{im}^{co}(1-w,d). \end{aligned}$$

As an example, consider the weight operator for conjunction based on the Kleene-Dienes implication:

$$g^{\wedge}(w,(s,d)) = (\max(1-w,s), \min(w,d)). \tag{27}$$

Consider the basic conjunction operator $\wedge$ for BSDs, which is defined by

$$\wedge : (\tilde{\mathbb{B}})^2 \to \tilde{\mathbb{B}} : ((s_1,d_1),(s_2,d_2)) \mapsto (\min(s_1,s_2), \max(d_1,d_2)) \tag{28}$$

(cf. Eq. (21)). Using this definition and the definition of the weight impact operator $g^{\wedge}$, a definition of an extended operator for weighted conjunction $\wedge^w$ of BSDs can now be given as follows:

$$\begin{aligned} \wedge^w : ([0,1] \times \tilde{\mathbb{B}})^2 &\to \tilde{\mathbb{B}} \\ ((w_1,(s_1,d_1)),(w_2,(s_2,d_2))) &\mapsto g^{\wedge}(w_1,(s_1,d_1)) \wedge g^{\wedge}(w_2,(s_1,d_1)). \end{aligned} \tag{29}$$

An extended operator for weighted disjunction can be defined analogously. Indeed, the impact of a weight on a BSD, in case of disjunction, can be defined by:

$$g^{\vee} : [0,1] \times \tilde{\mathbb{B}} \to \tilde{\mathbb{B}} : (w,(s,d)) \mapsto g^{\vee}(w,(s,d)) = \left(s_{g^{\vee}(w,(s,d))}, d_{g^{\vee}(w,(s,d))}\right) \tag{30}$$

where

$$\begin{aligned} s_{g^{\vee}(w,(s,d))} &= f_{im}^{co}(1-w,s) \\ d_{g^{\vee}(w,(s,d))} &= f_{im}(w,d). \end{aligned}$$

Using the Kleene-Dienes implication, the following weight operator for disjunction is for example obtained:

$$g^{\vee}(w,(s,d)) = (\min(w,s), \max(1-w,d)). \tag{31}$$

Consider the basic disjunction operator $\vee$ for BSDs, which is defined by

$$\vee : (\tilde{\mathbb{B}})^2 \to \tilde{\mathbb{B}} : ((s_1,d_1),(s_2,d_2)) \mapsto (\max(s_1,s_2), \min(d_1,d_2)) \tag{32}$$

(cf. Eq. (22)). Using this definition and the definition of the weight impact operator $g^{\vee}$, a definition of an extended operator for weighted conjunction $\vee^w$ of BSDs can then be given as follows:

$$\begin{gathered} \vee^w : ([0,1] \times \tilde{\mathbb{B}})^2 \to \tilde{\mathbb{B}} \\ ((w_1,(s_1,d_1)),(w_2,(s_2,d_2))) \mapsto g^{\vee}(w_1,(s_1,d_1)) \wedge g^{\vee}(w_2,(s_1,d_1)). \end{gathered} \tag{33}$$

Averaging

Besides the basic aggregation operators based on the aggregation of AFSs, using triangular norms and co-norms, BSDs can also be aggregated using other operators, like averaging operators [31]. Some averaging operators that could be used are the arithmetic mean (*AM*), geometric mean (*GM*) or harmonic mean (*HM*). As an example, consider the traditional arithmetic mean:

$$AM(x_1,\ldots,x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{34}$$

Such averaging operators cannot be applied on BSDs as such, because a BSD consists of a pair of values. So again, the satisfaction degrees and dissatisfaction degrees need to be treated separately. An extended version of the above regular averaging operator can be defined, where this regular averaging operator is applied for the satisfaction degrees, and, separately, for the dissatisfaction degrees:

$$AM((s_1,d_1),\ldots,(s_n,d_n)) = \left(\frac{1}{n}\sum_{i=1}^{n} s_i, \frac{1}{n}\sum_{i=1}^{n} d_i\right). \tag{35}$$

A similar extension can be defined for other averaging operators (*GM*, *HM*, ...).

Weighted Averaging

In the case of weighted averaging, it is again assumed that the importance of a condition, with respect to the final result, is linked with the condition itself, not with the degree to which the condition is satisfied. So weights $w_i \in [0,1]$ can be connected with the individual bipolar conditions. Again, in order to have an appropriate scaling, it is assumed that $\max_i w_i = 1$. Weighted counterparts of the above averaging operators for BSDs (e.g., weighted arithmetic mean (AM^w), weighted geometric mean (GM^w), or weighted harmonic mean (HM^w)) can be used, where the satis-

faction degrees on the one hand, and the dissatisfaction degrees on the other hand, are again aggregated separately using regular weighted averaging operators. As an example, consider the weighted arithmetic mean AM^w for BSDs:

$$\begin{aligned} AM^w &: ([0,1] \times \tilde{\mathbb{B}})^n \to \tilde{\mathbb{B}} \\ ((w_1,(s_1,d_1)),\ldots,(w_n,(s_n,d_n))) &\mapsto \left(\frac{\sum_{i=1}^n w_i \cdot s_i}{\sum_{i=1}^n w_i}, \frac{\sum_{i=1}^n w_i \cdot d_i}{\sum_{i=1}^n w_i} \right). \end{aligned} \tag{36}$$

4.2.2 Treating (s,d) as a Whole

Aggregating BSDs as a whole can be done by using Ordered Weighted Averaging (OWA) operators for BSDs [31]. Ordered weighted averaging of BSDs can be based on the traditional OWA operators [42, 43] as done in the case of aggregating regular satisfaction degrees. The OWA operator of dimension n, i.e., accepting n arguments $x_1,\ldots,x_n$ is defined by:

$$OWA_W(x_1,\ldots,x_n) = \sum_{i=1}^n w_i \cdot x'_i \tag{37}$$

where x'_i is the i^{th} largest value of $x_1,\ldots,x_n$ and $W = [w_1,\ldots,w_n]$; $\sum_{i=1}^n w_i = 1$ is a parameter of the OWA operator, referred to as the vector of weights.

This traditional OWA operator can also be extended to work with BSDs. To this aim, the BSDs are first rank ordered, for example by using one of the ranking functions presented in Section 3.2:

$$\begin{aligned} OWA_W &: \mathbb{B}^n \to \mathbb{B} \\ ((s_1,d_1),\ldots,(s_n,d_n)) &\mapsto \left(\sum_{i=1}^n w_i \cdot s'_i, \sum_{i=1}^n w_i \cdot d'_i \right) \end{aligned} \tag{38}$$

where (s'_i,d'_i) is the i^{th} largest BSD of $(s_1,d_1),\ldots,(s_n,d_n)$, according to the ranking function used.

Depending on the weight vector that is used, this extended OWA operator will behave differently (just like the regular OWA operator). In special cases, it can, e.g., act as a maximum function for BSDs ($w_1 = 1$, $w_i = 0$ for $i > 1$), a minimum function for BSDs ($w_n = 1$, $w_i = 0$ for $i < n$), or a median function for BSDs (for odd n: $w_{\lceil \frac{n}{2} \rceil} = 1$, $w_i = 0$ for $i \neq \lceil \frac{n}{2} \rceil$, where $\lceil\ \rceil$ denotes the ceiling function; for even n: $w_{\frac{n}{2}} = \frac{1}{2}$, $w_{\frac{n}{2}+1} = \frac{1}{2}$, $w_i = 0$ for $i \neq \frac{n}{2}$ and $i \neq \frac{n}{2}+1$).

Remark that the exact behaviour of the maximum, minimum and median function for BSDs (and also for all other OWA operators) depends on the specific ranking function employed.

4.3 Comparison and discussion

The aggregation of pairs of satisfaction degrees of elementary bipolar conditions should follow and reflect the semantics of the querying approach. This is why, for both approaches, specific aggregation techniques have been presented, hereby distinguishing techniques to aggregate both satisfaction degrees separately and to aggregate the satisfaction degree pairs as a whole.

In the constraint-wish approach, handling both satisfaction degrees separately boils down to treating all constraints together as a global constraint and treating all wishes together as a global wish, hereby preserving the applicable consistency condition, which requires some additional effort. Handling both satisfaction degrees as a whole boils down to lexicographical ordering. In both kinds of aggregation the semantics of constraints and wishes is retained.

In the satisfied-dissatisfied approach the satisfaction and dissatisfaction degrees are completely independent of each other. This characteristic offers more freedom to develop aggregation operators that treat both degrees separately.

- Basic aggregation operators, inspired by the aggregation of AFSs and based on the minimum triangular norm and maximum triangular co-norm, have been defined for non-weighted conjunction and disjunction. These operators retain the semantics of positive and negative information.
- To handle elementary query conditions of different importance, extended counterparts of the basic aggregation operators have been presented in literature. These operators use associated weights to model the relative importance of a query condition. First, the elementary conditions are evaluated as if there are no weights. Second, the impact of a weight on the evaluation of a condition is modelled in a premodification step using an implication and co-implication function.
- Instead of being based on a triangular norm and a triangular co-norm, an aggregation function can also be based on an averaging operator like the arithmetic, geometric or harmonic mean or on the weighted extension of such an averaging operator.

Choosing which aggregation operator to use depends on the requirements of the application. Aspects that may be considered in the selection of an adequate operator are: the need to better distinguish among the resulting records, the need for a reinforcement effect and the computation time.

BSDs can also be treated as a whole and aggregated based on their ranking. For that purpose, an OWA operator for BSDs has been presented in the literature. As is the case with regular OWA operators, the behaviour of the aggregation will then strongly depend on the used weight vector. Special cases are the minimum, maximum and median function for BSDs. Whether to use this kind of aggregation or not, and which weight vector should be chosen, again depend on the requirements of the application under consideration. Results obtained from an aggregation based on the ranking of BSDs are in general less informative to the user, because they do not provide independent information about the satisfaction of the positive and negative conditions in the user preferences. However, if a quantifier-based aggregation is

required by the application, where at least (or at most) a specified (fuzzy) number of elementary conditions should be satisfied in order to satisfy the query, an OWA-based aggregation can be used.

5 Conclusions

In this chapter, an overview and comparison of two commonly known approaches to bipolar querying of databases have been presented: the constraint-wish approach and the satisfied-dissatisfied approach. The specification of bipolar query conditions and different aspects of query handling, including the evaluation of elementary conditions, their aggregation, as well as ranking of the query results have been described.

The constraint-wish approach has been specifically designed to cope with situations where user preferences express requirements —called constraints— which should be satisfied (at least to some extent) by the retrieved database records, and other, optional conditions —called wishes— which serve to distinguish among those records that satisfy the constraints to the same extent. Slightly different semantics is modelled by the 'and possibly' based approach to constraints and wishes, where the influence of the wishes on the results of a query depends on the existence of the records satisfying constraints and wishes at the same time.

The motivation for the satisfied-dissatisfied approach is to cope with user preferences that are composed of positive conditions —expressing what the user likes— and negative conditions —expressing what the user wants to avoid. The positive and negative conditions do not necessarily have to be complementary to each other.

Although both approaches result in pairs of satisfaction degrees (constraint satisfaction and wish satisfaction, or satisfaction degree and dissatisfaction degree), the semantics are quite different. In the constraint-wish approach, 'true' constraints, i.e., mandatory requirements, are treated as more important in a specific sense. In the satisfied-dissatisfied approach, the positive and negative requirements are considered in general as being equally important and independent. Due to this assumed independence, it is also possible to model inconsistent or conflicting situations in the satisfied-dissatisfied approach, which is not possible in the constraint-wish approach, where either strong or weak consistency must apply. Moreover, in the satisfied-dissatisfied approach, the set of operators that can be used for ranking or aggregating is more elaborate than in the constraint-wish approach (e.g., weighted aggregation operators). On the other hand, a complete '*bipolar*' relational algebra has been proposed for the constraint-wish approach, i.e., an extension of traditional relational algebra to handle bipolarity [10].

References

1. Atanassov, K.: Intuitionistic fuzzy sets. Fuzzy Sets and Systems **20**, 87–96 (1986)
2. Belnap, N.D.: Modern Uses of Multiple-Valued Logic, chap. A useful four-valued logic, pp. 8–37. Reidel, Dordrecht, The Netherlands (1977)
3. Bordogna, G., Pasi, G.: Linguistic aggregation operators of selection criteria in fuzzy information retrieval. International Journal of Intelligent Systems **10**(2), 233–248 (1995)
4. Bosc, P., Kraft, D., Petry, F.: Fuzzy sets in database and information systems: Status and opportunities. Fuzzy Sets and Systems **156**, 418–426 (2005)
5. Bosc, P., Pivert, O.: Some approaches for relational databases flexible querying. International Journal of Intelligent Information Systems **1**, 323–354 (1992)
6. Bosc, P., Pivert, O.: An approach for a hierarchical aggregation of fuzzy predicates. In: Proc. of the 2nd IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93), pp. 1231–1236. San Francisco, USA (1993)
7. Bosc, P., Pivert, O.: Sqlf: a relational database language for fuzzy querying. IEEE Transactions on Fuzzy Systems **3**(1), 1–17 (1995)
8. Bosc, P., Pivert, O.: On three fuzzy connectives for flexible data retrieval and their axiomatization. In: Proc. of the SAC'11 Conference, pp. 1114–1118. Taiwan (2011)
9. Bosc, P., Pivert, O.: On four noncommutative fuzzy connectives and their axiomatization. Fuzzy Sets and Systems **202**, 42–60 (2012)
10. Bosc, P., Pivert, O., Mokhtari, A., Liétard, L.: Extending relational algebra to handle bipolarity. In: Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10), pp. 1718–1722. Sierre, Switzerland (2010)
11. De Calmès, M., Dubois, D., Hüllermeier, E., Prade, H., Sèdes, F.: A fuzzy set approach to flexible case-based querying: methodology and experimentation. In: Proc. of the 8th International Conference, Principles of Knowledge Representation and Reasoning (KR2002), pp. 449–458. Toulouse, France (2002)
12. De Tré, G., De Caluwe, R., Kacprzyk, J., Zadrożny, S.: On flexible querying via extensions to fuzzy sets. In: Proc. of the EUSFLAT'05 and LFA'05 Joint Conference, pp. 1225–1230. Barcelona, Spain (2005)
13. De Tré, G., Zadrożny, S., Matthé, T., Kacprzyk, J., Bronselaer, A.: Dealing with positive and negative query criteria in fuzzy database querying : bipolar satisfaction degrees. In: A. Troels (ed.) LECTURE NOTES IN COMPUTER SCIENCE, vol. 5822, pp. 593–604. Springer (2009)
14. Dubois, D., Prade, H.: Default reasoning and possibility theory. Artificial Intelligence **35**(2), 243–257 (1988)
15. Dubois, D., Prade, H.: Using fuzzy sets in flexible querying: why and how?, pp. 45–60. Kluwer Academic Publishers, Norwell, MA, USA (1997)
16. Dubois, D., Prade, H.: Bipolarity in flexible querying. Lecture Notes in Artificial Intelligence **2522**, 174–182 (2002)
17. Dubois, D., Prade, H.: Handbook of Research on Fuzzy Information Processing in Databases, chap. Handling bipolar queries in Fuzzy Information Processing, pp. 97–114. Information Science Reference, New York, USA (2008)
18. Dubois, D., Prade, H.: Gradualness, uncertainty and bipolarity: Making sense of fuzzy sets. Fuzzy Sets and Systems **192**, 3–24 (2012)
19. Grattan-Guinness, I.: Fuzzy membership mapped onto intervals and many-valued quantities. Mathematical Logic Quarterly **22**(1), 149–160 (1976)
20. Jahn, K.U.: Intervall-wertige mengen. Mathematische Nachrichten **68**(1), 115–132 (1975)
21. Kacprzyk, J., Zadrożny, S.: Fuzziness in database management systems, chap. FQUERY for Access: Fuzzy querying for windows-based DBMS, pp. 415–433. Physica-Verlag, Heidelberg, Germany (1995)
22. Kacprzyk, J., Zadrożny, S., Ziółkowski, A.: Fquery iii+: A "human-consistent" database querying system based on fuzzy logic with linguistic quantifiers. Information Systems **14**(6), 443–453 (1989)

23. Kacprzyk, J., Ziólkowski, A.: Database queries with fuzzy linguistic quantifiers. IEEE Transactions on Systems, Man and Cybernetics **16**, 474–479 (1986)
24. Konieczny, S., Marquis, P., Besnard, P.: Bipolarity in bilattice logics. International Journal of Intelligent Systems **23**(10), 1046–1061 (2008)
25. Lacroix, M., Lavency, P.: Preferences: Putting more knowledge into queries. In: Proc. of the VLDB'87 Conference, pp. 217–225. Brighton, UK (1987)
26. Liétard, L., Rocacher, D.: On the definition of extended norms and co-norms to aggregate fuzzy bipolar conditions. In: Proc. of the 2009 IFSA/EUSFLAT Conference, pp. 513–518. Lisbon, Portugal (2009)
27. Liétard, L., Rocacher, D., Bosc, P.: On the extension of sql to fuzzy bipolar conditions. In: Proc. of the 28th North American Information Processing Society Annual Conference (NAFIPS '09). Cincinnati, Ohio, USA (2009)
28. Liétard, L., Tamani, N., Rocacher, D.: Fuzzy bipolar conditions of type 'or else'. In: Proc. of the 2011 FUZZ-IEEE Conference, pp. 2546–2551. Taipei, Taiwan (2011)
29. Liétard, L., Tamani, N., Rocacher, D.: Linguistic quantifiers and bipolarity. In: Proc. of the 2011 IFSA World Congress and the 2011 AFSS International Conference. Surabaya and Bali Island, Indonesia (2011)
30. Matthé, T., De Tré, G.: Bipolar query satisfaction using satisfaction and dissatisfaction degrees: bipolar satisfaction degrees. In: Proc. of the ACM Symposium on Applied Computing (ACM SAC'09), pp. 1699–1703. Honolulu, Hawaii (2009)
31. Matthé, T., De Tré, G.: Weighted aggregation of bipolar satisfaction degrees. In: Proc. of the 2011 IFSA World Congress and the 2011 AFSS International Conference. Surabaya and Bali Island, Indonesia (2011)
32. Matthé, T., De Tré, G.: Ranking of bipolar satisfaction degrees. In: Proc. of the IPMU 2012 Conference, Communications in Computer and Information Sciences, Vol. 298, pp. 461–470. Catania, Italy (2012)
33. Matthé, T., De Tré, G., Zadrożny, S., Kacprzyk, J., Bronselaer, A.: Bipolar database querying using bipolar satisfaction degrees. International Journal of Intelligent Systems **26**(10), 890–910 (2011)
34. Öztürk, M., Tsoukiàs, A.: Modelling uncertain positive and negative reasons in decision aiding. Decision Support Systems **43**(4), 1512–1526 (2007)
35. Rivieccio, U.: Neutrosophic logics: Prospects and problems. Fuzzy Sets and Systems **159**(14), 1860–1868 (2008)
36. Sambuc, R.: Fonctions ϕ-floues. application à l'aide au diagnostic en pathologie thyroidienne. PhD thesis, Université de Marseille, France (1975)
37. Smarandache, F.: A Unifying Field in Logics: Neutrosophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability. American Research Press, Rehoboth, New Mexico, USA (1999)
38. Tahani, V.: A conceptual framework for fuzzy query processing: a step toward very intelligent database systems. Information Processing and Management **13**, 289–303 (1977)
39. Tamani, N., Liétard, L., Rocacher, D.: Bipolarity and the relational division. In: Proc. of the 7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-2011). Aix-les-Bains, France (2011)
40. Turunen, E., Öztürk, M., Tsoukiàs, A.: Paraconsistent semantics for Pavelka style fuzzy sentential logic. Fuzzy Sets and Systems **161**(14), 1926–1940 (2010)
41. Yager, R.: Fuzzy logic in the formulation of decision functions from linguistic specifications. Kybernetes **25**(4), 119–130 (1996)
42. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on Systems, Man and Cybernetics **18**(1), 183–190 (1988)
43. Yager, R.R., Kacprzyk, J.: The ordered weighted averaging operators : theory and applications. Kluwer Academic Publishers, Boston, USA (1997)
44. Zadeh, L.A.: Fuzzy sets. Information and Control **8**(3), 338–353 (1965)
45. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning - I. Information Sciences **8**(3), 199–249 (1975)

46. Zadrożny, S., De Tré, G., De Caluwe, R., Kacprzyk, J.: Handbook of Research on Fuzzy Information Processing in Databases, chap. An overview of fuzzy approaches to flexible database querying, pp. 34–54. Information Science Reference, New York, USA (2008)
47. Zadrożny, S., De Tré, G., Kacprzyk, J.: Remarks on various aspects of bipolarity in database querying. In: Proc. of the 2010 International Workshop on Database and Expert Systems Applications Proceedings (DEXA '10), pp. 323–327. Bilbao, Spain (2010)
48. Zadrożny, S., Kacprzyk, J.: Bipolar queries and queries with preferences. In: Proc. of the DEXA'06 Conference, pp. 415–419. Kraków, Poland (2006)