Modelling and solving bipolar preference problems

Stefano Bistarelli[1], Maria Silvia Pini[2], Francesca Rossi[2] and K. Brent Venable[2]

1 : University "G' D'Annunzio", Pescara, Italy
E-mail: bista@sci.unich.it
Istituto di Informatica e Telematica, CNR, Pisa, Italy
E-mail: Stefano.Bistarelli@iit.cnr.it
2 : Department of Pure and Applied Mathematics, University of Padova, Italy
E-mail: {mpini,frossi,kvenable}@math.unipd.it

Abstract. Real-life problems present several kinds of preferences. In this paper we focus on problems with both positive and negative preferences, that we call *bipolar problems*. Although seemingly specular notions, these two kinds of preferences should be dealt with differently to obtain the desired natural behaviour. We technically address this by generalizing the soft constraint formalism, which is able to model problems with one kind of preferences. We show that soft constraints model only negative preferences, and we define a new mathematical structure which allows to handle positive preferences as well. We also address the issue of the compensation between positive and negative preferences, studying the properties of this operation. Finally, we suggest how constraint propagation and branch and bound can be adapted to deal with bipolar problems.

1 Introduction

Many real-life problems contain statements which can be expressed as preferences. Moreover, preferences can be of many kinds: qualitative (as in "I like A more than B"), quantitative (as in "I like A at level 10 and B at level 11"), conditional (as in "If A happens, then I prefer B to C"), positive (as in "I like A, and I like B even more than A"), or negative (as in "I don't like A, and I really don't like B"). Our long-term goal is to define a framework where all such kinds of preferences can be naturally modelled and efficiently dealt. In this paper, we focus on problems which present positive and negative (quantitative and non-conditional) preferences, that we call *bipolar problems*.

Positive and negative preferences could be thought as two symmetric concepts, and thus one could think that they can be dealt with via the same operators and with the same properties. However, it is easy to see that this would not model what one usually expects in real scenarios. For example, when we have a scenario with two objects A and B, if we like both A and B, then the overall scenario should be more preferred than having just A or B alone. On the other hand, if we don't like A nor B, then the preference of the scenario should be smaller than the preferences of A or B alone. Thus combination of positive preferences should give us a higher (positive) preference, while combination of negative preferences should give us a lower (negative) preference.

When dealing with both kinds of preferences, it is natural to express also indifference, which means that we express neither a positive nor a negative preference over an object. For example, we may say that we like peaches, we don't like bananas, and we are indifferent to apples. A desired behaviour of indifference is that, when combined with any preference (either positive or negative), it should not influence the overall preference. For example, if we like peaches and we are indifferent to apples, a dish with peaches and apples should have overall a positive preference.

Finally, besides combining positive preferences among themselves, and also negative preferences among themselves, we also want to be able to combine positive with negative preferences. We strongly believe that the most natural and intuitive way to do so is to allow for compensation. Confronting positive against negative aspects and compensating them w.r.t. their strength is one of the core features of decision-making processes, and is, undoubtedly, a tactic universally applied to solve many real life problems. For example, if we have a meal with meat (which we like very much) and wine (which we don't like), then what should be the preference of the meal? To know that, we should be able to compensate the positive preference given to meat with the negative preference given to wine. The expected result is a preference which is between the two, and which should be positive if the positive preference is "stronger" than the negative one.

Soft constraints [3] are a useful formalism to model problems with quantitative preferences. However, they can model just one kind of preferences. In fact, we will see that technically they can model just negative preferences. Informally, the reason for this statement is that preference combination returns lower preferences, which, as mentioned above, is natural when using negative preferences. In this paper we start from the soft constraint formalism, based on c-semirings, to model negative preferences. We then extend it via a new structure, that models positive preferences, and we set the highest negative preference to coincide with the lower positive preference; this element models indifference. We then define a combination operator between positive and negative preferences to model preference compensation, and we study its properties in relation to the features of the preference structure. Finally, we consider the problem of finding optimal solutions of bipolar problems, by suggesting a possible adaptation of constraint propagation and branch and bound techniques to the generalized scenario. The structure we introduce to model both positive and negative preferences generalizes the one usually used for soft constraints. This allows for a natural and smooth extension of search and propagation algorithms for soft constraints to the bipolar setting.

Besides the possibility to use both kinds of preferences, which is the main aim of this paper, this generalization allows us also to use preference aggregation operators, such as the average or the median operators, which do not satisfy the properties required by the soft constraint formalism.

Parts of this paper have appeared in [4].

2 Background: semiring-based soft constraints

A soft constraint [3] is just a classical constraint [6] where each instantiation of its variables has an associated value from a (totally or partially ordered) set. This set has two operations, which makes it similar to a semiring, and is called a c-semiring. A c-semiring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that: A is a set and $\mathbf{0}, \mathbf{1} \in A$; $+$ is commutative, associative, idempotent, $\mathbf{0}$ is its unit element, and $\mathbf{1}$ is its absorbing element; $\times$ is associative, commutative, distributes over $+$, $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element.

Consider the relation $\leq_S$ over A such that $a \leq_S b$ iff $a+b=b$. Then: $\leq_S$ is a partial order; $+$ and $\times$ are monotone on $\leq_S$; $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum; $\langle A, \leq_S \rangle$ is a lattice and, for all $a, b \in A$, $a+b = lub(a, b)$. Moreover, if $\times$ is idempotent, then $\langle A, \leq_S \rangle$ is a distributive lattice and $\times$ is its glb. Informally, the relation $\leq_S$ gives us a way to compare (some of the) tuples of values and constraints. In fact, when we have $a \leq_S b$, we will say that *b is better than a*.

Given a c-semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$, a finite set D (the domain of the variables), and an ordered set of variables V, a constraint is a pair $\langle def, con \rangle$ where $con \subseteq V$ and $def : D^{|con|} \rightarrow A$. Therefore, a constraint specifies a set of variables (the ones in con), and assigns to each tuple of values of D of these variables an element of the semiring set A. Given a subset of variables $I \subseteq V$, and a soft constraint $c = \langle def, con \rangle$, the projection of c over I, written $c \Downarrow_I$, is a new soft constraint $\langle def', con' \rangle$, where $con' = con \cap I$ and $def(t') = \sum_{\{t | t\downarrow_{con'} = t'\}} def(t)$. In particular, the scope, con', of the projection constraint contains the variables that con and I have in common, and thus $con' \subseteq con$. Moreover, the preference associated to each assignment to the variables in con', denoted with t', is the highest ($\sum$ is the additive operator of the c-semiring) among the preferences associated by def to any completion of t', t, to an assignment to con.

A soft constraint satisfaction problem (SCSP) is just a set of soft constraints over a set of variables.

A classical CSP is just an SCSP where the chosen c-semiring is: $S_{CSP} = \langle \{false, true\}, \vee, \wedge, false, true \rangle$. On the other hand, fuzzy CSPs [10, 8] can be modeled in the SCSP framework by choosing the c-semiring: $S_{FCSP} = \langle [0, 1], max, min, 0, 1 \rangle$. For weighted CSPs, the semiring is $S_{WCSP} = \langle \Re^+, min, +, +\infty, 0 \rangle$. Here preferences are interpreted as costs from 0 to $+\infty$, which are combined with the sum and compared with min. Thus the optimization criterion is to minimize the sum of costs. For probabilistic CSPs [7], the semiring is $S_{PCSP} = \langle [0, 1], max, \times, 0, 1 \rangle$. Here preferences are interpreted as probabilities ranging from 0 to 1, which are combined using the product and compared using max. Thus the aim is to maximize the joint probability.

Given an assignment to all the variables of an SCSP, we can compute its preference value by combining the preferences associated by each constraint to the subtuples of the assignments referring to the variables of the constraint. For example, in fuzzy CSPs, the preference of a complete assignment t, written $pref(t)$, is the minimum preference given by the constraints. In weighted constraints, it is instead the sum of the costs given by the constraints. An optimal solution of an SCSP is then a complete assignment t such that there is no other complete assignment t'' with $pref(t) <_S pref(t'')$.

3 Negative preferences

The structure we use to model negative preferences is exactly a c-semiring, as defined in Section 2. In fact, in a c-semiring the element which acts as indifference is $\mathbf{1}$, since $\forall a \in A$, $a \times \mathbf{1} = a$. Notice that such element, denoted as $\mathbf{1}$ is not necessarily number 1 and in general it can be any element or number (0,1,100,X). This element is the best in the ordering, which is consistent with the fact that indifference is the best preference we can express when using only negative preferences.

Moreover, in a c-semiring, combination goes down in the ordering, since $a \times b \leq a, b$. This can be naturally interpreted as the fact that combining negative preferences worsens the overall preference.

This interpretation is very natural when considering, for example, the weighted semiring $(R^+, min, +, +\infty, 0)$. In fact, in this case the real numbers are costs and thus negative preferences. The sum of different costs is worse in general w.r.t. the ordering induced by the additive operator (that is, min) of the semiring.

Let us now consider the fuzzy semiring $([0, 1], max, min, 0, 1)$. According to this interpretation, giving a preference equal to 1 to a tuple means that there is nothing negative about such a tuple. Instead, giving a preference strictly less than 1 (e.g., 0.6) means that there is at least a constraint which such tuple doesn't satisfy at the best. Moreover, combining two fuzzy preferences means taking the minimum and thus the worst among them.

From now on, we will use a standard c-semiring to model negative preferences, denoted as: $(N, +_n, \times_n, \perp_n, \top_n)$.

4 Positive preferences

When dealing with positive preferences, we want two main properties to hold: combination should bring to better preferences, and indifference should be lower than all the other positive preferences. These properties can be found in the following structure.

Definition 1. *A positive preference structure is a tuple* $(P, +_p, \times_p, \perp_p, \top_p)$ *such that*

- P *is a set and* $\top_p, \perp_p \in P$*;*
- $+_p$*, the additive operator, is commutative, associative, idempotent, with* $\perp_p$ *as its unit element* ($\forall a \in P, a +_p \perp_p = a$) *and* $\top_p$ *as its absorbing element* ($\forall a \in P, a +_p \top_p = \top_p$)*;*
- $\times_p$*, the multiplicative operator, is associative, commutative and distributes over* $+_p$ ($a \times_p (b +_p c) = (a \times_p b) +_p (a \times_p c)$)*, with* $\perp_p$ *as its unit element and* $\top_p$ *as its absorbing element*[1].

Notice that the additive operator of this structure has the same properties as the corresponding one in c-semirings, and thus it induces a partial order over P in the usual way: $a \leq_p b$ iff $a +_p b = b$. This allows to prove that $+_p$ is monotone over $\leq_p$ and that it is the least upper bound in the lattice $(P, \leq_p)$.

On the other hand, the multiplicative operator has different properties. More precisely, the best element in the ordering ($\top_p$) is now its absorbing element, while the worst element ($\perp_p$) is its unit element. This reflects the desired behavior of the combination of positive preferences.

Theorem 1. *Given the positive preference structure* $(P, +_p, \times_p, \perp_p, \top_p)$*, consider the relation* $\leq_p$ *over* P*. Then:*

[1] The absorbing nature of $\top_p$ can be derived from the other properties.

– *$\times_p$ is monotone over $\leq_p$. That is, for any $a, b \in P$ such that $a \leq_p b$, then $a \times_p d \leq_p b \times_p d$, $\forall d \in P$.*
– *For any pair $a, b \in P$, $a \times_p b \geq_p a +_p b \geq_p a, b$.*

Proof. Since $a \leq_p b$ iff $a +_p b = b$, then $b \times_p d = (a +_p b) \times_p d = (a \times_p d) +_p (b \times_p d)$. Thus $a \times_p d \leq_p b \times_p d$. Also, $a \times_p b = a \times_p (b + \perp_p) = (a \times_p b) + (a \times_p \perp_p)$ $= (a \times_p b) + a$. Thus $a \times_p b \geq_p a$ (the same for b). Finally: $a \times_p b \geq a, b$. Thus $a \times_p b \geq lub(a, b) = a +_p b$. Q.E.D.

In a positive preference structure, $\perp_p$ is the element modelling indifference. In fact, it is the worst one in the ordering and it is the unit element for the combination operator $\times_p$. These are exactly the desired properties for indifference w.r.t. positive preferences.

The role of $\top_p$ is to model a very high preference, much higher than all the others. In fact, since it is the absorbing element of the combination operator, when we combine any positive preference a with $\top_p$, we get $\top_p$ and thus a disappears.

As a first example of a positive preference structure, consider $P_1 = (R^+, max, +, 0, +\infty)$, where preferences are positive reals. The smallest preference that can be assigned is 0. It represents the lack of any positive aspect and can thus be regarded as indifference. Preferences are aggregated taking the sum and are compared taking the max.

Another example is $P_2 = ([0, 1], max, max, 0, 1)$. In this case preferences are reals between 0 and 1, as in the fuzzy semiring for negative preferences. However, the combination operator is max, which gives, as a resulting preference, the highest one among all those combined.

As an example of a partially ordered positive preference structure consider the Cartesian product of the two described above: $(R^+ \times [0, 1], \langle max, max\rangle, \langle +, max\rangle, \langle 0, 0\rangle, \langle +\infty, 1\rangle)$. Positive preferences, here, are ordered pairs where the first element is a positive preference of type P_1 and the second one is a positive preference of type P_2. Consider for example the (incomparable) pairs $(8, 0.1)$ and $(3, 0.8)$. Applying the multiplicative operator will give pair $(11, 0.8)$ which, as expected, is better than both pairs since both $max(8, 3, 11) = 11$ and $max(0.1, 0.8, 0.8) = 0.8$.

5 Bipolar preference structures

Once we are given a positive and a negative preference structure, a first, naive, way to combine them is by performing the Cartesian product of the two structures. For example, if we have positive structure $(P, +_p, \times_p, \perp_p, \top_p)$ and negative structure $(N, +_n, \times_n, \perp_n, \top_n)$ the Cartesian product would be $(P \times N, \langle +_p, +_n\rangle, \langle \times_p, \times_n\rangle, \langle \perp_p, \perp_n\rangle, \langle \top_p, \top_n\rangle)$. In this setting, given a solution, it will be associated with a pair $\langle p, n\rangle$, where p is the overall positive preference and n is the overall negative preference. Such pair is an element of the carrier of the new structure. Clearly, the new structure is not a positive nor a negative preference structure, and, in fact, some pairs will be neither clearly positive nor negative. The ordering induced over the pairs is the well known Pareto ordering, which declares as incomparable any two solutions defeating each other on one component. Although simple, this criterion is not satisfactory in practice since it may induce a lot of incomparability among the solutions. This drawback can be traced to the inability of compensating positive and negative preferences. Such ability is, instead, one of the key features of another, more sophisticated, bipolar structure which we will now describe.

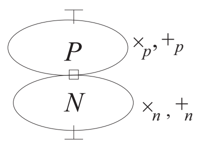
Definition 2. *A bipolar preference structure is a tuple* $(N, P, +, \times, \bot, \Box, \top)$ *where*

- $(P, +_{|P}, \times_{|P}, \Box, \top)$ *is a positive preference structure;*
- $(N, +_{|N}, \times_{|N}, \bot, \Box)$ *is a c-semiring;*
- $+ : (N \cup P)^2 \longrightarrow (N \cup P)$ *is such that* $a_n + a_p = a_p$ *for any* $a_n \in N$ *and* $a_p \in P$*; this operator induces as partial ordering on* $N \cup P$*:* $\forall a, b \in P \cup N$, $a \leq b$ *iff* $a + b = b$*;*
- $\times : (N \cup P)^2 \longrightarrow (N \cup P)$ *is an operator (called the* compensation operator*) that, for all* $a, b, c \in N \cup P$*, satisfies the following properties:*
  - *commutativity:* $a \times b = b \times a$*;*
  - *monotonicity: if* $a \leq b$*, then* $a \times c \leq b \times c$*.*

In the following, we will write $+_n$ instead of $+_{|N}$ and $+_p$ instead of $+_{|P}$. Similarly for $\times_n$ and $\times_p$. Moreover, we will sometimes write $\times_{np}$ when operator $\times$ will be applied to a pair in $(N \times P)$.

Bipolar preference structures generalize c-semirings. In fact, a c-semiring is just a bipolar preference structure with a single positive preference: the indifference element, which, in such a case, is also the top element of the structure. Similarly, bipolar preference structures generalize positive structures. In fact, the latter are just bipolar preference structures with a single negative preference: the indifference element. By symmetry, in such cases the indifference element coincides with the bottom element of the structure.

Given the way the ordering is induced by $+$ on $N \cup P$, easily, we have $\bot \leq \Box \leq \top$. Thus, there is a unique maximum element (that is, $\top$), a unique minimum element (that is, $\bot$); the element $\Box$ is smaller than any positive preference and greater than any negative preference, and it is used to model indifference. The shape of a bipolar preference structure is shown in the following figure:



Despite the ordering suggested by the figure, which places all positive preferences strictly above negative preferences, our framework does not prevent from using the same scale to represent both positive and negative preferences. Such a case can be easily handled by using two isomorphisms: one between an instance of the scale and the positive preference structure, and another one between another instance of the same scale and the negative preference structure. The same holds also when one wishes to use partially overlapping scales.

A bipolar preference structure allows us to have different ways to model and reason about positive and negative preferences. In fact, we can have different lattices $(P, \leq_P)$ and $(N, \leq_n)$. For example, we can have a richer structure for one kind of preference.

This is common in real-life problems, where negative and positive statements are not necessarily expressed using the same granularity. For example, we could be satisfied with just two levels of negative preferences, while requiring ten levels of positive preferences. Nevertheless, our framework allows to model cases in which the two structures are isomorphic, as well.

It is easy to show that the combination of a positive and a negative preference is a preference which is higher than, or equal to, the negative one and lower than, or equal to, the positive one. The following theorems hold when a bipolar preference structure $(N, P, +, \times, \bot, \Box, \top)$ is given.

Theorem 2. *For all* $p \in P$ *and* $n \in N$*,* $n \leq p \times n \leq p$*.*

Proof. For any $n \in N$ and $p \in P$, $\Box \leq p$ and $n \leq \Box$. By monotonicity of $\times$, we have: $n \times \Box \leq n \times p$ and $n \times p \leq \Box \times p$. Hence: $n = n \times \Box \leq n \times p \leq \Box \times p = p$. Q.E.D.

This means that the compensation of positive and negative preferences must lie in one of the chains between the two combined preferences. Notice that all such chains pass through the indifference element $\Box$. Possible choices for combining strictly positive with strictly negative preferences are thus the average or the median operator.

Moreover, by monotonicity, we can show that if $\top \times \bot = \bot$, then the result of the compensation between any positive preference and the bottom element is the bottom element, and if $\top \times \bot = \top$, then the compensation between any negative preference and the top element is the top element.

Theorem 3. *Given bipolar preference structure* $(N, P, +, \times, \bot, \Box, \top)$*:*

- *if* $\top \times \bot = \bot$*, then* $\forall p \in P$*,* $p \times \bot = \bot$*;*
- *if* $\top \times \bot = \top$*, then* $\forall n \in N$*,* $n \times \top = \top$*.*

Proof. Assume $\top \times \bot = \bot$. Since for all $p \in P$, $p \leq \top$, then, by monotonicity of $\times$, $p \times \bot \leq \top \times \bot = \bot$, hence $p \times \bot = \bot$.

Assume $\top \times \bot = \top$. Since for all $n \in N$, $\bot \leq n$, then, by monotonicity of $\times$, $\top = \top \times \bot \leq \top \times n$, hence $\top \times n = \top$. Q.E.D

A bipolar structure may satisfy the following (additional) property:

$$[P1]: \ \forall p \in P', \exists n \in N' \ s.t. \ p \times n = \Box \ and \ viceversa \tag{1}$$

where,

- if $\top \times \bot = \Box$, $P' = P$ and $N' = N$,
- if $\top \times \bot = p \in P - \{\Box\}$, $P' = P - \{\top\}$ and $N' = N$,
- if $\top \times \bot = n \in N - \{\Box\}$, $P' = P$ and $N' = N - \{\bot\}$.

In Property $P1$, if $\top \times \bot = p \in P - \{\Box\}$, then $P' = P - \{\top\}$, since in this case there is no element that combined with $\top$ produces the indifference element. In fact, $\forall n \in N$, $\bot \leq n$, and so, by monotonicity of $\times$, $\Box < p = \top \times \bot \leq \top \times n$. Analogously, if $\top \times \bot = n \in N - \{\Box\}$, then $N' = N - \{\bot\}$, since there is no element that combined with $\bot$ produces the indifference element.

5.1 Examples of bipolar preference structures

In the following table each row corresponds to a bipolar preference structure.

N,P	$+_p, \times_p$	$+_n, \times_n$	$\times_{np}$	$\bot, \Box, \top$
R^-, R^+	max, sum	max, sum	sum	$-\infty$, 0, $+\infty$
$[-1, 0]$, $[0, 1]$	max, max	max, min	sum	-1, 0, 1
$[0, 1]$, $[1, +\infty]$	max, prod	max, prod	prod	0, 1, $+\infty$

The structure described in the first row uses positive real numbers as positive preferences and negative reals as negative preferences. Compensation is obtained by summing the preferences, while the ordering is given by the max operator. In the second structure we have positive preferences between 0 and 1 and negative preferences between -1 and 0. The compensation between positive preferences is max, between negative preferences is min and between positive and negative preferences is sum and the order is given by max. In the third structure we use positive preferences between 1 and $+\infty$ and negative preferences between 0 and 1. Compensation is obtained by multiplying the preferences and ordering is again via max. If $\top \times \bot \in \{\top, \bot\}$, then compensation in the first and in the third structure is associative.

5.2 Associativity of preference compensation

In general, the compensation operator $\times$ may be not associative. Here we list some sufficient conditions for the non-associativity of the $\times$ operator.

Theorem 4. *Given a bipolar preference structure* $(P, N, +, \times, \bot, \Box, \top)$*, if*

- $\top \times \bot = c \in (N \cup P) - \{\top, \bot\}$*;*
- *or* $\exists p \in P - \{\top\}$ *and* $n \in N - \{\bot\}$ *such that* $p \times n = \Box$ *and at least one of the following conditions holds:*
  - $\times_p$ *or* $\times_n$ *is idempotent;*
  - $\exists p' \in P - \{p, \top\}$ *such that* $p' \times n = \Box$ *or* $\exists n' \in N - \{n, \bot\}$ *such that* $p \times n' = \Box$*;*
  - $\top \times \bot = \bot$ *and* $\exists n' \in N - \{\bot\}$ *such that* $n \times n' = \bot$*;*
  - $\top \times \bot = \top$ *and* $\exists p' \in P - \{\top\}$ *such that* $p \times p' = \top$*;*
  - $\exists a, c \in N \cup P$ *such that* $a \times p = c$ *iff* $c \times n \neq a$ *(or* $\exists a, c \in N \cup P$ *such that* $a \times n = c$ *iff* $c \times p \neq a$*),*

then operator $\times$ *is not associative:*

Proof.
- If $c \in P - \{\top\}$, then $\top \times (\top \times \bot) = \top \times c = \top$, while $(\top \times \top) \times \bot = \top \times \bot = c$. If $c \in N - \{\bot\}$, then $\bot \times (\bot \times \top) = \bot \times c = \bot$, while $(\bot \times \bot) \times \top = \bot \times \top = c$.
- Assume that $\exists p \in P - \{\top\}$ and $n \in N - \{\bot\}$ such that $p \times n = \Box$.
  - If $\times_p$ is idempotent, then $p \times (p \times n) = p \times \Box = p$, while $(p \times p) \times n = p \times n = \Box$. Similarly if $\times_n$ is idempotent.
  - If $\exists p' \in P - \{p, \top\}$ such that $p' \times n = \Box$, then $(p \times n) \times p' = p'$, while $p \times (n \times p') = p$. Analogously, if $\exists n' \in N - \{n, \bot\}$ such that $p \times n' = \Box$.

- If $\top \times \bot = \bot$, then, by Theorem 3, $p \times \bot = \bot$. If $\exists n' \in N - \{\bot\}$ such that $n \times n' = \bot$, then $(p \times n) \times n' = \Box \times n' = n'$, while $p \times (n \times n') = p \times \bot = \bot \neq n'$.
- If $\top \times \bot = \top$, then, by Theorem 3, $n \times \top = \top$. If $\exists p' \in P - \{\top\}$ such that $p \times p' = \top$, then $(n \times p) \times p' = \Box \times p' = p'$, while $n \times (p \times p') = n \times \top = \top \neq p'$.
- If $c \times n \neq a$, then $(a \times p) \times n = c \times n \neq a$, but $a \times (p \times n) = a \times \Box = a$. Analogously if $c \times p \neq a$.

Q.E.D.

Notice that these sufficient conditions refer to various aspects of a bipolar preference structure: properties of the operators, shape of the orderings of P and N, the relation between $\times$ and the other operators. Since some of these conditions often occur in practice, it is not reasonable to require associativity of $\times$.

For example, $\times$ is not associative when the combination between $\top$ and $\bot$ is different from $\top$ or $\bot$, or when the combination operator of either the positive or the negative preferences is idempotent. This result depends on the fact that the proposed framework allows to choose the result of the compensation between $\top$ and $\bot$, and the operators $\times_n$ and $\times_p$, as long as the monotonicity of $\times$ is respected.

In Theorem 4 it is shown that if either $\times_p$ or $\times_n$ is idempotent, then $\times$ is not associative. However, there are also cases in which both $\times_p$ and $\times_n$ are not idempotent, and still $\times$ is not associative. For example, this happens when there are two different preferences that combined with the same preference give the indifference element. Another sufficient condition for the non-associativity of the compensation operator concerns the presence of at least two negative (resp. positive) preferences different from $\bot$ (resp. $\top$), such that their combination is $\bot$ (resp. $\top$). Consider, for example, a bipolar preference structure where N=[-50,0], P=[0,100], $+=$ max, $\times =$ bounded-sum, $\bot = -50, \Box = 0$, and $\top = 100$. In this case, preferences such as 50 and 60 are not equal to the top (100) but their bounded sum obtains 100. As expected, $-10 + (50 + 60) = -10 + 100 = 90$, while $(-10 + 50) + 60 = 40 + 60 = 100$. Another case that lead to non associativity of $\times$ is when there are two preference values that don't behave like inverse elements in ordinary algebra.

6 Bipolar preference problems

Once we have defined bipolar preference structures, we can define a notion of bipolar constraint, which is just a constraint where each assignment of values to its variables is associated to one of the elements in a bipolar preference structure.

Definition 3. *Given a bipolar preference structure* $(N, P, +, \times, \bot, \Box, \top)$*, a finite set* D *(the domain of the variables), and an ordered set of variables* V*, a constraint is a pair* $\langle def, con \rangle$ *where* $con \subseteq V$ *and* $def : D^{|con|} \rightarrow (N \cup P)$.

A bipolar CSP (V, C) is then just a set of variables V and a set of bipolar constraints C over V.

There could be many ways of defining the optimal solutions of a bipolar CSP. Here we propose a simple one which compensates only preferences of complete instantiations. This avoids problems due to the possible non-associativity of the compensation operator, since compensation never involves more than two preference values. Thus the preference of a solution does not depend on the order in which the preferences of its constraints are aggregated.

Definition 4. *A solution of a bipolar CSP (V, C) is a complete assignment to all variables in V, say s, and an associated preference which is computed as follows: $pref(s) = (p_1 \times_p \ldots \times_p p_k) \times (n_1 \times_n \ldots \times_n n_l)$, where $p_i \in P$ for $i := 1, \ldots, k$ and $n_j \in N$ for $j := 1, \ldots, l$ and $\exists \langle def, con \rangle \in C$ such that $p_i = def(s \downarrow_{con})$ or $n_j = def(s \downarrow_{con})$. A solution s is an optimal solution if there is no other solution s' with $pref(s') > pref(s)$.*

In this definition, the preference of a solution s is obtained by combining all the positive preferences associated to its projections over the constraints, by using $\times_p$, combining all the negative preferences associated to its projections over the constraints, by using $\times_n$, and then, combining the two preferences obtained so far (one positive and one negative) by using the operator $\times_{np}$.

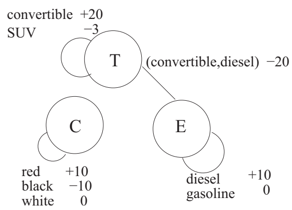
If $\times$ is associative, then other definitions of solution preference could be used while giving the same result. In fact, any combination of aggregation and compensation, applied to the preferences of the constraints of the problem, would lead to the same overall preference, and thus to the same solution ordering.

6.1 An example of bipolar CSP

Consider the scenario in which we want to buy a car. We have some preferences over the car's features. In terms of color, we like red, we are indifferent to white, and we hate black. Also, we like convertible cars a lot and we don't care much for SUVs. In terms of engines, we like diesel. However, we don't want a diesel convertible.

We may decide to represent positive preferences via positive integers and negative preferences via negative integers. Moreover, we may decide to maximize the sum of all kinds of preferences. This can be modelled by a preference structure where $N = [-\infty, 0]$, $P = [0, +\infty]$, $+$ =max, $\times$=sum, $\bot = -\infty$, $\Box = 0$, $\top = +\infty$.

We have three variables: variable T (type) with domain {convertible, SUV}, variable E (engine) with domain {diesel, gasoline}, and variable C (color) with domain {red, white, black}. For the preferences over the colors, we define a constraint $c_1 = \langle def_1, \{C\} \rangle$ where, for example, we set def_1(red) $= +10$, def_1(black) $= -10$, and def_1(white) $= 0$. We also have a constraint over car types, say $c_2 = \langle def_2, \{T\} \rangle$ where we set def_2(convertible) $= +20$ and def_2(SUV) $= -3$. The constraint over engines can then be $c_3 = \langle def_3, \{E\} \rangle$, where we can set def_3(diesel) $= +10$ and def_3(gasoline) $= 0$. Finally, the last preference can be modelled by a constraint $c_4 = \langle def_4, \{T, E\} \rangle$, where we can set def_4(convertible, diesel) $= -20$ and $def_4(a, b) = 0$ for $(a, b) \neq$(convertible, diesel). The following figure shows the structure of such a bipolar CSP.



Notice that we have set the preference values in a way that models the intuitive strength of the preferences described informally in the example. Moreover, we have used the value 0 to model indifference.

Consider, now, solution s_1 =(red,convertible,diesel): $pref(s_1) = (def_1(\text{red}) \times def_2(\text{convertible}) \times def_3(\text{diesel})) \times def_4(\text{convertible, diesel}) = (10 + 20 + 10) + (-20) = 20$. Analogously, we can compute the preference of all other solutions and see that the optimal solution is (red, convertible, gasoline) with global preference of 30.

Consider now a different bipolar preference structure, which differs from the previous one only for $\times_p$, which is now max. Now solution s_1 has preference $pref(s_1) = (def_1(\text{red}) \times def_2(\text{convertible}) \times def_3(\text{diesel})) \times def_4(\text{convertible, diesel}) = max(10, 20, 10) + (-20) = 0$. It is easy to see that now an optimal solution has preference 20. There are two of such solutions: one is the same as the optimal solution above, and the other one is (white,convertible, gasoline). The two cars have the same features except for the color. A white convertible is just as good as a red convertible because we decided to aggregate positive preference by taking the maximum elements rather than by summing them.

6.2 Solving bipolar CSPs

Bipolar problems are NP-hard, since they generalize both classical and soft constraints, which are already known to be difficult problems [3]. Preference problems based on c-semirings can be solved via a branch and bound technique, possibly augmented via soft constraint propagation, which may lower the preferences and thus allow for the computation of better bounds [3].

In bipolar CSPs, we have both positive and negative preferences. We propose to use an algorithm similar to Branch and Bound algorithm (BnB) [6] used for unipolar preferences. Being able to do so is a good point since it allows to handle bipolar preferences without much additional effort.

Following BnB, whenever a solution is found, its preference, if higher that those found before, is kept as a lower bound, L, for the optimal preference in the maximization task. Moreover, for each partial solution t an upper bound, $ub(t)$, is computed by overestimating the best preference of a solution extending t. If $ub(t) \leq L$, i.e. the preference of the best solution in the subtree below t is worse than the preference of the best solution found so far, then the subtree below t is pruned.

Our algorithm is different from standard BnB in that it allows the compensation operator to be non-associative. This may require to consider some total completions of t in order to compute $ub(t)$.

More precisely, we adapt BnB to compute, at each search node k corresponding to a partial assignment t, an upper bound to the preferences of all the solutions in the k-rooted subtree as follows.

- If $\times$ is not associative, then each node is associated to a positive and a negative preference, say p and n, which are obtained by aggregating all preferences of the same type obtained in the instantiated part of the problem. Next all the best preferences (which may be positive or negative) in the uninstantiated part of the problem are considered. By aggregating those of the same type, we get a positive and a negative preference, say p' and n', which can be combined with the ones associated to the current node. This produces the following upper bound $ub = (p \times_p p') \times (n \times_n n')$, where $p' = p_1 \times_p \ldots \times_p p_w$, $n' = n_1 \times_n \ldots \times_n n_s$, with $w + s = r$, where r is the number of uninstantiated variables/constraints. Hence ub can be computed via $r - 1$ aggregation steps and one compensation step.
- If $\times$ is associative, then we don't need to postpone compensation until all constraints have been considered. This means that we can keep just one preference value for each search node, $v = p \times n$, that can be positive or negative, which is obtained by aggregating all preferences (both positive and negative) obtained in the instantiated part of the problem. The same can be done considering the best preferences in the uninstantiated part of the problem, obtaining a value v'. Thus, ub can now be written as $ub = v \times v'$, where $v' = a_1 \times \ldots \times a_r$, where $a_i \in N \cup P$ is the best preference found in a constraint of the uninstantiated part of the problem. Thus now ub can be computed via at most $r - 1$ steps among which there can be many compensation steps. A compensation can generate the indifference element $\Box$, which is the unit element for the compensation operator. Thus, when $\Box$ is generated, the successive computation step can be avoided.

Algorithm 1 shows the pseudocode of the procedure we propose to compute the upper bound within the BnB algorithm. The input is a partial assignment t to a subset $X = \{x_1 \ldots, x_k\}$ of the set of variables $V = \{x_1, \ldots x_n\}$ and the bipolar CSP, P', obtained from the initial bipolar CSP by reducing the domains of the variables in X to the sigleton corresponding to their assignment in t.

For every constraint $c =< def, con >\in C$, constraint $c \Downarrow_{X,t}$ obtained by projecting c on X and considering only the subtuple $t \downarrow_{X \cap con}$ is considered. We will denote $c \Downarrow_{X,t}$ with c' and we will denote with C' the union set of all such constraints. Note that, by the definition of projection constraint (Section 2), $c \Downarrow_{X,t}$ associates to subtuple $t \downarrow_{X \cap con}$ the best preference associated by def to any of its completions to variables in con.

If $\times$ is not associative, then the algorithm computes the aggregation $p(t)$ of all the best preferences that are positive, i.e., the preferences obtained on each constraint $c^+ \in C'$ such that $def_{c^+}(t \downarrow_{X \cap con_{c^+}}) \in P$ and the aggregation $n(t)$ of all the best preferences that are negative, i.e. the preferences obtained on each constraints $c^- \in C'$ such that $def_{c^-}(t \downarrow_{X \cap con_{c^-}}) \in N$. The final step compensates between $p(t)$ and $n(t)$ and returns the result, $ub(t)$, of this compensation.

If $\times$ is associative then the algorithm aggregates directly the best preferences that can be positive or negative and it returns the result of this aggregation, i.e. $ub(t)$.

If $\times_n$ is idempotent, then, to improve this upper bound, we can propagate negative preferences as it is done in soft constraints [3, 5]. In fact, such a propagation may lower

Algorithm 1: Upper Bound computation

Input: t: assignment to variables in $X = \{x_1, \ldots, x_k\}$
P': bipolar CSP;
Output: ub(t): preference;
foreach $c \in C$ **do**
  compute $c' = c \Downarrow_{X,t}$
$C' \leftarrow \cup_{c \in C} c'$;
if $\times$ *is not associative* **then**
  $p(t) \leftarrow \prod_{p\ \{c^+ \in C'\}} def_{c^+}(t \downarrow_{con_{c^+}})$;
  $n(t) \leftarrow \prod_{n\ \{c^- \in C'\}} def_{c^-}(t \downarrow_{con_{c^-}})$;
  $ub(t) \leftarrow p(t) \times n(t)$
else
  $ub(t) \leftarrow \prod_{\{c' \in C'\}} def_{c'}(t \downarrow_{con_{c'}})$;
return $ub(t)$;

the negative values while not changing the semantics of the problem. Due to monotonicity of $\times$ and $\times_n$, the upper bound may become smaller and allow for more pruning. Notice that, if $\times_n$ is idempotent, then $\times$ cannot be associative (see Theorem 4). Hence, we can perform propagation only when $\times$ is not associative and $\times_n$ idempotent. In this case, since compensation among positive and negative preferences is performed only at the end (when all the variables are instantiated), only negative preferences associated to domain values can be lowered through propagation.

Propagation can be achieved by a standard Arc Consistency (AC) algorithm for soft constraints [6] with an adapted *Revise* function. For simplicity, we assume to have only binary soft constraints and we will denote with c_{ij} the soft constraint defined on variables x_i and x_j. As usual, AC starts initializing a queue, containing pairs of variables, adding pair (x_i, x_j) and pair (x_j, x_i) for every constraint c_{ij} in the problem. While the queue is not empty, a pair, (x_i, x_j) is popped from the queue and function *Revise* is applied to it. If *Revise* causes some changes (i.e., it lowers the preference of some value of the domain of x_i) then all pairs containing x_i, but not x_j, are pushed in the queue. The *Revise* function is shown in Algorithm 2.

Revise takes in input a pair of variables (x_i, x_j), the domains of the variables, D_{x_i} and D_{x_j} with the functions, def_i and def_j, associating preferences to the values in the domains, and the soft constraint c_{ij}. The output is the domain of the first variable of the pair, D_{x_i}, with a possibly revised preference function associating lower preferences to some values of the domain. The preference associated to an element, t_i, in the domain of x_i is changed as described above. If $\times$ is not associative, the revision occurs only if the preference associated to t_i is negative ($def_i(t_i) \in N$), the highest preference associated to any tuple of c_{ij} in which $x_i = t_i$ is negative ($\sum_{t_j \in D_{x_j}} def_{ij}(t_i t_j) \in N$) and all the preferences associated to elements in the domain of x_j are negative ($def_j(t_j) \in N, \forall t_j \in D(x_j)$).

Algorithm 2: Revise

Input: pair of variables (x_i, x_j),
soft constraint $c_{ij} =< \{x_i, x_j\}, def_{ij} >$,
Domain of x_i, D_{x_i} and preference function $def_i : D_{x_i} \rightarrow N \cup P$,
Domain of x_j, D_{x_j} and preference function $def_j : D_{x_j} \rightarrow N \cup P$
Output: Domain D_{x_i}, and function def_i arc-consistent w.r.t. x_j
foreach $t_i \in D_{x_i}$ **do**
  if *($\times$ not associative and $\times_n$ idempotent* **and** $def_i(t_i) \in N$ **and** $def_j(t_j) \in N, \forall t_j \in D_{x_j}$ **and** $\sum_{t_j \in D_{x_j}} def_{ij}(t_i t_j) \in N$*)* **then**
    $p(t_i) \leftarrow def_i(t_i) \times \sum_{t_j \in D_{x_j}} (def_{ij}(t_i t_j) \times def_j(t_j))$
    if $p(t_i) < def_i(t_i)$ **then**
      $def_i(t_i) \leftarrow p(t_i)$;

7 Related work

Bipolar reasoning and preferences have recently attracted interest in the AI community.

In [5], fair preference structures are introduced. In such a structure, which is an ordered set with an operation, $\oplus$, the key concept is that of difference of two elements. In particular, a structure is said to be fair if for each pair of ordered elements, $\alpha \leq \beta$, there exists a maximal element, γ, such that $\alpha \oplus \gamma = \beta$ called the difference of β and α. Although there is some similarity with the behaviour of our compensation operator, in [5], the setting is unipolar and the goal is mainly algorithmic (extension of arc consistency to Valued CSPs), rather than concerned with modelling new types of preferences.

In [1, 2] a bipolar preference model based on a fuzzy-possibilistic approach is described. The main differences with the framework presented in this paper are the fact that only fuzzy preferences are considered and that negative preferences are interpreted as violations of constraints. In particular, the approach followed to combine negative and positive preferences in [1, 2] is that of giving precedence to the negative preference optimization and resorting to positive preferences only to distinguish among the optimals found in the first step. Positive and negative preferences are, thus, kept separate and no compensation is allowed.

In [9] the authors consider totally ordered unipolar and bipolar preference scales. In this paper we present a method to deal with partially ordered bipolar scales. When the preference set is totally ordered, operators $\times_n$ and $\times_p$ described here correspond respectively to the $t - norm$ and $t - conorm$ used in [9]. Moreover, in [9] an operator, the $uninorm$, similar to the compensation operator but with the restriction of always being associative is considered. Due to the associativity requirement, our compensation operator is more general and may not be a uninorm when restricted to totally ordered scales.

8 Future work

We plan to develop a solver for bipolar CSPs, which should be flexible enough to accommodate for both associative and non-associative compensation operators. We also intend to consider the presence of uncertainty in bipolar problems, possibly using possibility theory and to develop solving techniques for such scenarios. Another line of future research is the generalization of other preference formalisms, such as multicriteria methods and CP-nets, to deal with bipolar preferences and to study the relation between bipolarization and importance tradeoffs. Finally, we plan to consider the possible connections between our work and non-monotonic concurrent constraints.

References

1. S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Bipolar representation and fusion of preferences in the possibilistic logic framework. In *KR 2002*. Morgan Kaufmann, 2002.
2. S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Bipolar possibility theory in prefrence modeling: representation, fusion and optimal solutions. *Information Fusion, an International Journal on Multi-Sensor, Multi-Source Information Fusion*, 2006.
3. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint solving and optimization. *Journal of the ACM*, 44(2):201–236, mar 1997.
4. S. Bistarelli, M. S. Pini, F. Rossi, and K. B. Venable. Bipolar preference problems. In *ECAI-06 (poster)*, 2006.
5. M. Cooper and T. Schiex. Arc consistency for soft constraints. *AI Journal*, 154(1-2):199–227, 2004.
6. R. Dechter. *Constraint processing*. Morgan Kaufmann, 2003.
7. H. Fargier and J. Lang. Uncertainty in constraint satisfaction problems: a probabilistic approach. In *ECSQARU 93*, volume 747 of *LNCS*. Springer, 1993.
8. H. Fargier, T. Schiex, and G. Verfaille. Valued Constraint Satisfaction Problems: Hard and Easy Problems. In *IJCAI-95*, pages 631–637. Morgan Kaufmann, 1995.
9. M. Grabisch, B. de Baets, and J. Fodor. The quest for rings on bipolar scales. *Int. Journ. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2003.
10. Zs. Ruttkay. Fuzzy constraint satisfaction. In *3rd IEEE International Conference on Fuzzy Systems*, pages 1263–1268, 1994.