Soft Constraint Problems With Uncontrollable Variables

Maria Silvia Pini MPINI@MATH.UNIPD.IT
Francesca Rossi FROSSI@MATH.UNIPD.IT
Corresponding author: Kristen Brent Venable KVENABLE@MATH.UNIPD.IT
Department of Pure and Applied Mathematics,
Via Trieste, 63 - 35121 PADOVA
Tel: +39 049 8271200 Fax: +39 049 8271499
University of Padova, Italy

Keywords:Constraint programming, preferences, uncertainty, possibility theory.

Abstract

Preferences and uncertainty are common in many real-life problems. In this paper we consider preferences modelled via soft constraints, that allow for the representation of quantitative preferences. Moreover, we consider uncertainty modelled via uncontrollable variables, that is, variables whose value cannot be decided by us. We also assume that some information is provided for such variables, in the form of a possibility distribution over their domains. Possibilities provide a way to model imprecise probabilities, and give us a way to know which values are more possible than others for the uncontrollable variables. For such problems, the aim is to find a solution with a high preference which is also very robust with respect to the uncontrollable part.

To tackle such problems, we adopt an existing approach that eliminates the uncertain part of the problem while adding some constraints in the remaining part, and then solves the resulting problem. However, contrarily to the specific methods present in the literature, we formulate several desirable properties, on the robustness of the problem's solutions and its relationship with their preferences, that should be satisfied by any specific method based on this approach. We also define several semantics to order the solutions according to different attitudes to risk, and we discuss which of the desirable properties are satisfied by each of the considered semantics. Finally, we present a solver for this kind of problems, and we show some experimental results of its application over classes of such problems.

1. Introduction

Preferences and uncertainty are present in many application fields, such as scheduling, logistics, and production planning. For example, in a satellite scheduling problem, the activities of a satellite taking pictures of a part of the earth should be scheduled according to some preferences on the energy consumption and the quality of the pictures, and also considering the uncertainty on the weather, since clouds may hide the part to be photographed. Also, in a production planning problem, a plan has to be decided by following certain optimization criteria, starting from a description of the initial world, of the available decisions and their effects, and of the goals to reach, but also by taking into account possible unexpected events during the production. In a temporal scheduling problem, there are usually constraints and preferences over the order of execution of various activities, and the durations of some activities may be uncertain (Dubois, Fargier, & Prade, 1995). In this case the goal is to define a temporal schedule of the activities which is highly preferred and also robust with respect to the uncertain durations. Finally, in energy trading and network traffic analysis (Yorke-Smith &

Gervet, 2003) uncertainty is due to incomplete or erroneous data information. In energy trading costs may be imprecise since they may evolve due to market changes; in network traffic analysis the overwhelming amount of information and measurement difficulties force the use of partial or imprecise information. The goal in general is to provide robust solutions to the problem, i.e., solutions that hold under the maximum number of possible states of the world (Fargier, Lang, & Schiex, 1996) or by reasoning upon probabilistic data distributions (Walsh, 2002). Another approach is to reason about uncertain problems by modelling explicitly what is known about uncertain data in terms of an uncertain constraint satisfaction problem and to compute its full closure, i.e., the set of the possible solutions to the model or a subset of it as the user specifies (Yorke-Smith & Gervet, 2003).

Preferences can be of many kinds. For example, they can be qualitative or quantitative, bipolar or unipolar, conditional or not, etc. Quantitative preferences associate a preference value to an event, while qualitative ones usually give a relation among the events to state the preference ordering over them. Examples of preference representation formalisms are CP-nets (Boutilier, Brafman, Hoos, & Poole, 1999; Boutilier, Brafman, Domshlak, Hoos, & Poole, 2004), soft constraints (Bistarelli, Montanari, & Rossi, 1997), and utility theory (Fishburn, 1979).

Uncertainty can also come in several guises, and has been handled in different ways. Probabilities are often used, and, when precise probabilities are not available, also possibility theory (Zadeh, 1999; Dubois & Prade, 1988) has been adopted. Possibility theory basically provides an upper and a lower bound to probabilities. Indeed, possibilistic uncertainty is useful in many domains, such as data analysis (Wolkenhauer, 1998; Tanaka & Guo, 1999), structural learning (Borgelt, Gebhardt, & Kruse, 2000), database querying (Bosc & Prade, 1997), diagnosis (Cayrac, Dubois, Haziza, & Prade, 1996), belief revision (Benferhat, Dubois, Prade, & Williams, 2002), argumentation (Amgoud & Prade, 2004), and case-based reasoning (Huellermeier, 2007). Moreover, experimental results in cognitive psychology (Raufaste, Neves, & Marin, 2003) suggest that there are situations where people reason about uncertainty using the rules of possibility theory, rather than those of probability theory.

In this paper, we consider problems with quantitative preferences and possibilistic uncertainty. This kind of problems occurs often in real life, since quantitative preferences model costs, utilities, as well as hard requirements, and since often we don't have precise information about the uncontrollable part of the problem.

More precisely, we describe real-life problems via a set of variables with finite domains and a set of soft constraints among subsets of the variables. Soft constraints allow us to express preferences over the instantiations of the variables of the constraints. We will consider in particular *fuzzy preferences*, that employ values between 0 and 1, and that are combined using the min operator. This means that each event is given a value between 0 and 1, with higher values being associated to more preferred events, and that a conjoint event is given a preference value which is the smallest of the preference levels of the single events. Thus we only remember the worst event when several events occur. Fuzzy preferences are very useful in many application domains, such as the spatial (Dominguez & Klinko, 2004) or the medical one (A. Blinowska, 1993), where it is often necessary to have a pessimistic approach.

We model uncertainty in such problems by the presence of *uncontrollable* variables. This means that the value of such variables will not be decided by us, but by Nature or by some other agent. Thus a solution will not be an assignment to all the variables, but only to the controllable ones. Although we cannot choose the value for such uncontrollable variables, usually we have some information on the plausibility of the values in their domains. In (Fargier, Lang, Martin-Clouaire, & Schiex, 1995a) this information is given in terms of probability distributions. In this paper, we model this information by a *possibility distribution* (Zadeh, 1999) over the values in the domains of such variables.

As mentioned before, possibilities are useful when probability distributions are not available, and provide an upper and a lower bound to probabilities (Zadeh, 1999). Moreover, they allow for easy integration with fuzzy preferences (Dubois, Fargier, & Prade, 1996), since both possibilities and fuzzy preferences associate values between 0 and 1 to events.

For the solutions of this kind of problems (that is, fuzzy constraints plus uncontrollable variables with a possibility distribution), we define the notion of preference and robustness. We also discuss and define several desirable properties that such notions should respect, also in relation to the solution preference ordering.

To find the optimal solutions of such problems, we eliminate the uncontrollable variables (and all the constraints connecting them) and we add new fuzzy constraints in the controllable part. Such new constraints contain, in a different form, some of the information that was present in the removed part. While the old constraints are used to compute the preference of a solution, these additional constraints are used to compute its robustness.

We then introduce several semantics that use the notions of preference and robustness to order the solutions. For all such semantics, and also for one present in the literature (Dubois et al., 1996), we study if the notions of preference and robustness, when one of such semantics is used, satisfy the desired properties. The semantics introduced in this paper appear to be a finer way to evaluate the solutions of these problems (with respect to the existing literature), since they allow us to distinguish between highly preferred solutions which are not robust, and robust but not preferred solutions. For example, they guarantee that, if there are two solutions with the same robustness (resp., the same preference), then the ordering is given by their preference (resp., robustness).

In the last part of the paper we describe a solver we have developed to solve fuzzy constraint problems with possibilistic uncertainty, that uses the approach just described to solve a problem of this kind: first it eliminates the uncertain part, and then it solves a fuzzy constraint problem with the additional robustness information. The solution of the fuzzy constraint problem is obtained by exploiting branch and bound techniques and constraint propagation (Rossi, Van Beek, & Walsh, 2006).

The paper is structured as follows. In Section 2 we present the basic notions about soft constraints, fuzzy constraints, and possibility theory. In Section 3 we define the class of problems we consider. In Section 4 we discuss the notions of preference and robustness, as well as some desirable properties of such notions. In Section 5 we define our method to remove uncertainty from the problems. Based on this method, in Section 6 we define formally the notions of preference and robustness, and we show that the notion of robustness satisfies the desired properties. In Section 7 we propose various semantics for ordering the solutions with respect to their preference and robustness values. We also show that the solution orderings that such semantics induce satisfy the desired properties. In Section 8.3 we present in detail an example of a problem with fuzzy constraints and uncertainty, and of the application of our procedure. In Section 8 we present our solver, and in Section 10 we test the solver over several problem instances with specific features. Finally, in Sections 11 and 12 we discuss some related work, we summarize our results, and we point out possible directions for future research.

All the proofs of the results in this paper are contained in the appendix. Preliminary versions of some parts of this paper have appeared in (Pini, Rossi, & Venable, 2005) and (Pini & Rossi, 2005).

2. Background

In this section we give an overview of the background on which our work is based. First, we present a formalism for representing soft preferences, i.e., the semiring-based soft constraints (Bistarelli et al., 1997). Then, we describe the formalism we will use for representing uncertainty, i.e., possibility theory (Zadeh, 1999).

2.1 Soft constraints

In the literature there are many formalizations of the concept of soft constraints (Schiex, Fargier, & Verfaillie, 1995; Ruttkay, 1994). Here we refer to the one described in (Bistarelli et al., 1997; Bistarelli, Montanari, & Rossi, 1995), which however can be shown to generalize and express many others (Bistarelli, Fargier, Montanari, Rossi, Schiex, & Verfaillie, 1996).

In a few words, a soft constraint (Bistarelli et al., 1997) is just a classical constraint (Dechter, 2003) where each instantiation of its variables has an associated value from a (totally or partially ordered) set. Combining constraints will then have to take into account such additional values, and thus the formalism has also to provide suitable operations for combination ($\times$) and comparison ($+$) of tuples of values and constraints. This is why this formalization is based on the concept of c-semiring, which is just a set plus two operations. More precisely, a c-semiring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$[1] such that: A is a set, called the carrier of the c-semiring, 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 $a, b \in A$, $a + b = lub(a, b)$ where lub indicates the least upper bound operator. Moreover, if $\times$ is idempotent, then $\langle A, \leq_S\rangle$ is a distributive lattice and $\times$ is its greatest lower bound operator. 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*. Thus, $\mathbf{0}$ is the worst value and $\mathbf{1}$ is the best one.

Definition 1 (soft constraints) 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 in D of these variables an element of the semiring set A. This element can be interpreted in many ways: as a level of preference, or as a cost, or as a probability, etc. The correct way to interpret such elements determines the choice of the semiring operations.

Definition 2 (SCSP) A soft constraint satisfaction problem, denoted by $\langle S, V, C\rangle$ is a set of soft constraints C based on the c-semiring S, which is defined over a set of variables V. □

[1] when $+$ (respectively $\times$) is applied to a two-element set we will use symbol $+$ (respectively $\times$) in infix notation, while in general we will use the the symbol $\sum$ (respectively $\prod$) in prefix notation.

Definition 3 (combination) Given two constraints $c_1 = \langle def_1, con_1 \rangle$ and $c_2 = \langle def_2, con_2 \rangle$, their *combination* $c_1 \otimes c_2$ is the constraint $\langle def, con \rangle$, where $con = con_1 \cup con_2$ and $def(t) = def_1(t \downarrow^{con}_{con_1}) \times def_2(t \downarrow^{con}_{con_2})$[2]. □

The combination operator $\otimes$ can be straightforwardly extended also to finite sets of constraints: when applied to a finite set of constraints C, we will write $\bigotimes C$. In words, combining constraints means building a new constraint involving all the variables of the original ones, and which associates to each tuple of domain values for such variables a semiring element, which is obtained by multiplying the elements associated by the original constraints to the appropriate subtuples.

Definition 4 (projection) Given a constraint $c = \langle def, con \rangle$ and a subset I of V, the *projection* of c over I, written $c \Downarrow_I$, is the constraint $\langle def', con' \rangle$ where $con' = con \cap I$ and $def'(t') = \sum_{t/t \downarrow^{con}_{I \cap con} = t'} def(t)$. □

Informally, projecting means eliminating some variables. This is done by associating to each tuple over the remaining variables a semiring element, which is obtained by summing (via the operator $+$ of the semiring) the elements associated by the original constraint to all the extensions of this tuple over the eliminated variables.

Definition 5 (solution) A solution of a SCSP $\langle S, C, V \rangle$ is a complete instantiation, $(d_1, \ldots, d_n)$, of the variables in $V = \{x_1, \ldots x_n\}$. □

Each solution has a preference value, obtained by combining, via the $\times$ operator, the preference levels of its subtuples given in the various constraints.

Definition 6 (solution preference) Given SCSP $\langle S, C, V \rangle$ and a solution s, the preference of s is $pref(s) = \Pi_{\langle def_i, con_i \rangle \in C\}} def_i(s \downarrow_{con_i})$. □

Definition 7 (optimal solution) Given a SCSP P and a solution s, s is optimal if and only if $\nexists\, s'$, solution of P, such that $pref(s') >_S pref(s)$. □

SCSPs can be solved by extending and adapting the techniques usually used for classical CSPs. For example, to find the best solution, we could employ a branch-and-bound search algorithm (instead of the classical backtracking). Also the so-called constraint propagation techniques, like arc-consistency (Rossi et al., 2006) and path-consistency, can be generalized to SCSPs (Bistarelli et al., 1995, 1997).

Choosing a specific semiring means selecting a class of preferences. For example, the semiring $S_{FCSP} = \langle [0,1], max, min, 0, 1 \rangle$ allows one to model Fuzzy CSPs (FCSPs) (Ruttkay, 1994; Schiex, 1992), which associate to each element allowed by a constraint a preference between 0 and 1 (with 0 being the worst and 1 being the best preference), and gives to each complete assignment a preference that is the minimal among all preferences selected in the constraints. The optimal solutions are then those solutions with the maximal preference.

[2] By $t \downarrow^X_Y$ we mean the projection of tuple t, which is defined over the set of variables X, over the subset of variables $Y \subseteq X$.

2.2 Possibility theory

In this paper we will consider SCSPs with uncertainty, where uncertainty is modelled by the presence of uncontrollable variables. This means that the value for such variables will not be decided by us, but the Nature or by some other agent. Although we cannot choose the values for such variables we assume, in this paper, to have some information on the plausibility of the values in their domain. We model this information by possibility distributions over the values in the domains of such variables.

Possibility distributions were defined in the possibility theory, introduced in (Zadeh, 1999), in connection with fuzzy set theory (Zadeh, 1999; Dubois & Prade, 1980, 2000), to allow reasoning on vague knowledge. This theory is useful to deal with uncertain events, when probabilistic estimations are not available, for instance, because there is no similar events to be referred to.

More formally, a possibility distribution π_x associated to an uncontrollable variable x represents the set of more or less plausible and mutually exclusive values of x.

Definition 8 (possibility distribution) A possibility distribution π_x associated to a single valued variable x with domain D is a mapping from D to a totally ordered scale L (usually $[0, 1]$) such that $\forall d \in D$, $\pi_x(d) \in L$ and $\exists\ d \in D$ with $\pi_x(d) = 1$, where 1 the top element of the scale L. □

In particular, $\pi_x(d) = 0$ means $x = d$ is impossible, while $\pi_x(d) = 1$ means $x = d$ is fully possible. A possibility distribution is similar to a probability density. However, $\pi_x(d) = 1$ only means that $x = d$ is a plausible situation, which cannot be excluded. Thus, a degree of possibility can be viewed as an upper bound of a degree of probability. Possibility theory encodes incomplete knowledge, while probability accounts for random and observed phenomena. In particular, the possibility distribution π_x can encode *complete ignorance* about x: $\pi_x(d) = 1, \forall d \in D$. In this case all values $d \in D$ are plausible for x, and so it is impossible to exclude any of them. Moreover, it can encode *complete knowledge* about x: $\pi_x(\bar{d}) = 1, \exists \bar{d} \in D$ and $\pi_x(d) = 0, \forall d \in D$ s.t. $d \neq \bar{d}$. In this case only the value $\bar{d}$ is plausible for x.

Given a possibility distribution π_x associated to a variable x, the occurrence of the event $x \in E \subseteq D$ can be defined by the possibility and the necessity degrees.

Definition 9 (possibility degree) The possibility degree of an event "$x \in E$", denoted by $\Pi(x \in E)$, or simply by $\Pi(E)$, is $\Pi(x \in E) = max_{d \in E} \pi_x(d)$. □

The possibility degree of the event "$x \in E$' evaluates the extent to which "$x \in E$" is *possibly* true. In particular, $\Pi(x \in E) = 1$ means that the event $x \in E$ is totally possible. However it could also not happen. Therefore in this case we are completely ignorant about its occurrence. $\Pi(x \in E) = 0$ means that the event $x \in E$ for sure will not happen.

Definition 10 (necessity degree) The necessity degree of "$x \in E$", denoted by $N(x \in E)$ or simply by $N(E)$, is $N(x \in E) = min_{d \in \bar{E}} r(\pi_x(d))$, where r is the order reversing map in the interval $[0, 1]$ such that $\forall p \in [0, 1]$, $r(p) = 1 - p$, and $\bar{E}$ is the complement of E in D. □

The necessity degree of the event "$x \in E$" evaluates the extent to which "$x \in E$" is *certainly* true. In particular, $N(x \in E) = 1$ means that the event $x \in E$ is certain, while $N(x \in E) = 0$ means that the event is not necessary at all, although it may happen. In fact, $N(x \in E) = 0$ if and only if $\Pi(x \in \bar{E}) = 1$.

The possibility and the necessity degrees of an event $x \in E$ are respectively, an upper and a lower bound of the probability degree, say $P(E)$, of that event (i.e., $N(E) \leq P(E) \leq \Pi(E)$). Moreover, the possibility and the necessity measures are related by the following formula $\Pi(E) = 1 - N(\bar{E})$. From this, follows $N(E) = 1 - \Pi(\bar{E})$.

3. Uncertain soft constraint problems

Uncertain soft constraint problems (USCSPs) are SCSPs where some variables are uncontrollable, i.e., they are not under the user's control. They can model many real-life problems, such as scheduling and timetabling. For example, they can model the problem of scheduling some tasks, knowing that the duration of some of those is uncertain, and only vaguely known (Dubois et al., 1995), or the problem of deciding how many training sessions to perform in a tutorial, without knowing the effective number of participants, but knowing only an approximately number of these participants (Dubois et al., 1996).

Contrarily to classical constraint problems, in USCSPs we cannot decide how to assign the variables to make the assignment optimal, but we must assign values to the controllable variables, denoted with V_c, guessing what Nature will do with the uncontrollable variables, denoted with V_u. Depending on the assumptions made on the observability of the uncontrollable variables, different optimality criteria can be defined.

For example, an optimal solution for an USCSP can be defined as an assignment of values to the variables in V_c such that, *whatever* Nature will decide for variables in V_u, the overall assignment will be optimal. This corresponds to assume that the values of the uncontrollable variables are never observable, i.e., that the values of the controllable variables are decided upon without observing the values of the uncontrollable variables. This is a pessimistic view, and often an assignment satisfying such a requirement doesn't exist. In this case one can relax the optimality condition to that of having a preference above a certain threshold α in all scenarios. Hence solving the problem will consist of finding those assignments to variables in V_c which satisfy this property at the highest α. Furthermore, one could be satisfied with finding an assignment of values to the variables in V_c such that, *for at least* one assignment decided by Nature for the variables in V_u, the overall assignment will be optimal. This definition follows an optimistic view. Other definitions can be between these two extremes.

If the uncontrollable variables are equipped with additional information on the likelihood of their values, like in our case, such an information can be used to infer new soft constraints over the controllable variables, which express the compatibility of the controllable part of the problem with the uncontrollable one. As we will show in the next section, this information can be used to define the notion of optimal solution.

In this paper, we want to reason on uncertain problems by assuming to have no observability over uncertain events before decision. We define an USCSP as a set of variables, which can be controllable or uncontrollable, and a set of soft constraints over these variables. Moreover, we assume the domain of every uncontrollable variable is equipped with a possibility distribution, that specifies, for every value in the domain, the degree of plausibility that the variable takes that value. Formally,

Definition 11 (USCSP) An uncertain soft CSP is a tuple $\langle S, V_c, V_u, \pi, C_c, C_{cu}, C_u \rangle$, where

- S is a c-semiring:

- $V_c = \{x_1, \dots x_n\}$ is a set of controllable variables;
- $V_u = \{z_1, \dots z_k\}$ is a set of uncontrollable variables
- $\pi = \{p_1, \dots p_k\}$ is a set of possibility distributions over V_u. In particular, every $z_i \in V_u$ has possibility distribution π_i with scale $[0, 1]$;
- C_c is the set of constraints that involve only variables of V_c;
- C_{cu} is a set of constraints that involve at least a variable in V_c and a variable in V_u, and that may involve any other variable of $(V_c \cup V_u)$;
- C_u is the set of constraints that involve only variables of V_u. □

Notice that when the set of uncontrollable variables, i.e., V_u, of an USCSP $\langle S, V_c, V_u, \pi\ C_c, C_{cu}, C_u\rangle$, is empty, then the sets of constraints involving variables in V_u, i.e., C_{cu} and C_u, are empty, and the USCSP corresponds to an SCSP $\langle S, V_c, C_c\rangle$, as defined in Definition 2.

When the chosen semiring is $S_{FCSP} = \langle [0, 1], max, min, 0, 1\rangle$, Definition 11 models an Uncertain Fuzzy CSP (UFCSP), that corresponds, when there are no uncontrollable variables, to an FCSP, as defined in Section 2.1.

Example 1 Figure 1 shows an example of an UFCSP. Each constraint is defined by associating a preference level (in this case between 0 and 1) to each assignment of its variables to values in their domains. The set V_c of the controllable variables is composed by x, y, and w, while the set V_u of the uncontrollable variables contains only z. The values in the domain of z are characterized by the possibility distribution π_Z. The set of constraints C_c is composed by the constraint $\langle q, \{x, w\}\rangle$, which relates x and w via the preference function q. The set of constraints C_{cu} is composed by the constraint $\langle f, \{x, y, z\}\rangle$, which is defined on variables x, y, and z by the preference function f, while the set C_u is empty. □

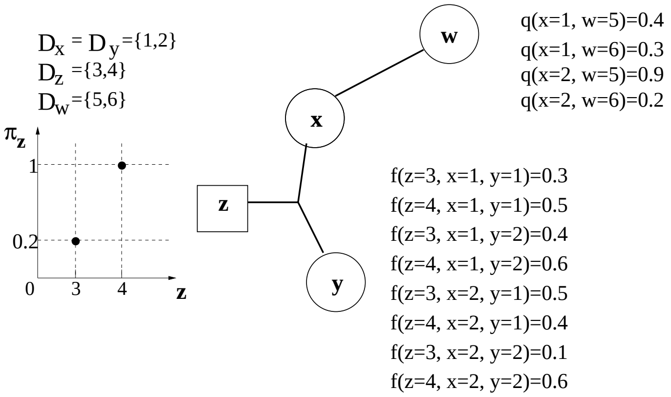


Figure 1: An uncertain fuzzy CSP.

Given an assignment t to all the variables of an USCSP, its overall preference is computed by combining, via the × operator, the preference levels of its subtuples in the selected constraints.

Definition 12 (overall assignment preference) Given an USCSP $Q = \langle S, V_c, V_u, \pi, C_c, C_{cu}, C_u \rangle$, let t be an assignment to all the variables of Q, then its preference is the value $ovpref(t) = \prod_{\{\langle def_i, con_i \rangle \in C_c \cup C_{cu} \cup C_u\}} def_i(t \downarrow_{con_i})$. □

A solution of an USCSP is a complete assignment to all its controllable variables. More formally,

Definition 13 (solution) Given an USCSP $Q = \langle S, V_c, V_u, \pi, C_c, C_{cu}, C_u \rangle$, a solution of Q is a complete assignment to all the variables of V_c. □

In what follows we will assume to have USCSPs with $C_u = \emptyset$. If $C_u \neq \emptyset$, we can translate every constraint of type C_u in a new constraint of type C_{cu}, thus obtaining a USCSP with $C_u = \emptyset$. This procedure will be described in Section 8.1.

4. Preference, robustness and desirable properties

Given a solution s of a USCSP, we will associate a preference degree to it, written $pref(s)$, which summarizes all the preferences in the controllable part, and that could be obtained for some assignment to the uncontrollable variables decided by Nature. It is reasonable to assume that $pref(s)$ belongs to the set of preferences in the considered c-semiring.

When we deal with USCSPs, we have to consider another interesting aspect that characterizes a solution, that is, its robustness with respect to the uncertainty. In general terms, the robustness of a solution measures the impact of Nature on the preference obtained by choosing that solution. There are several ways to define the robustness of a solution. An intuitive measure of robustness of a solution s is the one which considers the worst preference which can be obtained for s in the constraints connecting controllable and uncontrollable variables. However, if we have possibility distributions over the values in the domains of uncontrollable variables, it is easy to imagine that the robustness of s should depend both on the preferences in the constraints connecting both controllable and uncontrollable variables to s, and on such possibility distributions. It is also reasonable that the robustness of a solution should be an element of the c-semiring. This will allow us to use the operators of such a structure over the robustness values.

Before giving our definition of robustness of a solution s, that we will denote with $rob(s)$, we define two properties that such a definition should satisfy.

Property 1 *Given solutions s and s' of an USCSP $\langle S, V_c, V_u, \pi, C_c, C_{cu}, \emptyset \rangle$, where every variable v_i in V_u is associated to a possibility distribution π_i, if, for every constraint $\langle def, con \rangle \in C_{cu}$, and for every assignment a to the uncontrollable variables in con, $def((s, a) \downarrow_{con}) \leq_S def((s', a) \downarrow_{con})$, then it should be that $rob(s) \leq_S rob(s')$.*

In other words, if we increase the preference of tuples involving uncontrollable variables, we should get a higher value of robustness.

Property 2 *Take a solution s of the USCSPs $Q_1 = \langle S, V_c, V_u, \pi_1, C_c, C_{cu}, \emptyset \rangle$, $Q_2 = \langle S, V_c, V_u, \pi_2, C_c, C_{cu}, \emptyset \rangle$. Assume that for every assignment a to variables in V_u, $\pi_2(a) \leq \pi_1(a)$. Then it should be that $rob_1(s) \leq_S rob_2(s)$, where rob_1 is the robustness computed in the problem Q_1, and rob_2 is the robustness computed in the problem Q_2.*

In other words, if we lower the possibility of values of uncontrollable variables, we should get a higher value of robustness.

To understand which solutions are better than others in a USCSP, it is reasonable to consider a solution ordering, say $\succ$, which should be reflexive and transitive. No matter if it is a partial or a total order, the notions of robustness and preference should be related to this solution ordering by the following properties.

Property 3 *Get two solutions s and s' of an USCSP $Q = \langle S, V_c, V_u, \pi, C_c, C_{cu}, \emptyset \rangle$. If $ovpref(s, a) >_S ovpref(s', a)$, $\forall a$ assignment to V_u, then it should be that $s \succ s'$.*

In other words, if two solutions s and s' are such that the overall preference of the assignment (s, a) to all the variables is better than or equal to one of (s', a) for all the values a of the uncontrollable variables, then s should be considered better than or equal to s'.

Property 4 *Given two solutions s and s' of an USCSP, if $rob(s) = rob(s')$ and $pref(s) >_S pref(s')$, it should be that $s \succ s'$.*

Property 5 *Given two solutions s and s' of an USCSP, if $pref(s) = pref(s')$, and $rob(s) >_S rob(s')$, then it should be that $s \succ s'$.*

In other words, two solutions which are equally good with respect to one aspect (robustness or preference degree) and differ on the other one should be ordered according to the discriminating aspect.

5. Removing uncertainty from UFCSPs

While in the previous sections we have considered USCSPs, where any semiring could be chosen, we now focus on uncertain fuzzy CSPs (UFCSPs) to make the treatment simpler and easier to understand. We now propose a procedure that eliminates uncontrollable variables from UFCSPs while preserving as much information as possible. Starting from this procedure, we define the preference and the robustness degrees of a solution, and we show that such notions satisfy the desirable properties mentioned above.

This procedure, that we call Algorithm *SP* (for *separation* and *projection*), is applied in the solver for UFCSPs, that we will describe in detail in the next sections, only after having performed a preprocessing phase and a propagation phase. The preprocessing phase connects every uncontrollable variable not connected with a controllable one, to at least a controllable variable, by adding new induced constraints. The propagation phase, instead, propagates all the constraints, including the new ones. These two steps are performed in the solver before applying *SP* in order to mitigate the loss of information that can arise after having applied *SP*, since it removes uncontrollable variables without taking into account the information contained in the constraints involving only uncontrollable variables and since it processes separately the constraints involving controllable and uncontrollable variables even if they share some variables.

Algorithm *SP* takes as input an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, C_u \rangle$, where $C_u = \emptyset$, and returns a structure (that we call an RFCSP) which is similar to an FCSP but has two sets of constraints rather than one.

Definition 14 (RFCSP) An RFCSP is a tuple $\langle S_{FCSP}, V_c, C_1, C_2 \rangle$ such that $\langle S_{FCSP}, V_c, C \rangle$, where $C = C_1 \cup C_2$, is an FCSP. □

The RFCSP Q' returned by *SP* is obtained from the UFCSP Q by eliminating its uncontrollable variables and the fuzzy constraints in C_{cu}, and by adding new sets of fuzzy constraints, i.e., C_{rob} and C_{proj}, only among the controllable variables. In particular, it adds C_{proj} to C_c, while it keeps C_{rob} separate. More precisely, C_{proj} (*the projection constraints*) is the set of constraints obtained applying to every constraint c in C_{cu} of Q the procedure $ComputeProjectionConstraint(c)$, that will be described in Section 5.2, while C_{rob} (*the robustness constraints*) is the set of constraints obtained applying to every constraint c in C_{cu} of Q the procedure $ComputeRobustnessConstraint(c)$, that will be described in Section 5.1. As mentioned above, both projection and robustness constraints are not sufficient to prevent the loss of information occurring when the uncontrollable part of the problem is deleted. In Sections 5.1 and 5.2 we will see in detail how these new constraints encode some of the information contained in the uncontrollable part of the problem.

Starting from this problem Q', we then define the preference degree of a solution considering the preference functions of the constraints in $C_c \cup C_{proj}$, and the robustness degree of a solution considering the preference functions of the constraints in C_{rob}.

Algorithm 1: *SP*

```
Input: $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset\rangle$: an USCSP;
Output: $Q' = \langle S_{FCSP}, V_c, C_c^*, C_{rob}\rangle$: an RFCSP;
$C_{rob} \leftarrow \emptyset$;
$C_{proj} \leftarrow \emptyset$;
foreach constraint $c \in C_{cu}$ do
    $C_{rob} \leftarrow C_{rob} \cup ComputeRobustnessConstraint(c)$;
    $C_{proj} \leftarrow C_{proj} \cup ComputeProjectionConstraint(c)$;
$C_c^* \leftarrow C_c \cup C_{proj}$;
$Q' \leftarrow \langle S_{FCSP}, V_c, C_c^*, C_{rob}\rangle$;
return $Q'$;
```

5.1 Robustness constraints

The set of robustness constraints C_{rob} is obtained by reasoning on the preference functions of the constraints in C_{cu} and on the possibilities associated to values in the domains of uncontrollable variables involved in such constraints.

Every constraint in C_{rob} is built by exploiting the procedure described in (Dubois et al., 1996). Such a procedure, denoted $ComputeRobustnessConstraint$ in algorithm *SP*, takes a constraint $c = \langle def, con\rangle \in C_{cu}$, relating controllable and uncontrollable variables, such that $con \cap V_u = Z$ and $con \cap V_c = X$, and it returns a fuzzy constraint $c' = \langle def', con'\rangle$ such that $con' = con \cap V_c$. The preference function of any assignment t_X to X is

$$def'(t_X) = N(t_X\ satisfies\ c) = min_{t_Z \in A_z} max(def(t_X, t_Z), r(\pi_Z(t_Z))),$$

where t_Z is an assignment to Z, A_Z is the domain of Z, and r is the order reversing map such that $r(p) = 1 - p$, $\forall p \in [0, 1]$. In words, the preference value $def'(t_X)$ is given by the necessity degree that the assignment t_X partially satisfies the constraint c, i.e., by the necessity degree that the preference $def(t_X, t_Z)$ is greater than 0 when varying t_Z in c (Dubois et al., 1996).

This value, that has the form of the pessimistic possibilistic expected utility (Dubois & Prade, 1995; Dubois, Godo, Prade, & Zapico, 1998; Dubois et al., 1996), is characterized by the following property, that we will denote with **Property PO** (Dubois et al., 1996):

$$def'(t_X) \geq \alpha \quad \textit{if and only if} \quad def(t_X, t_Z) \geq \alpha, \forall t_Z \textit{ with } \pi_Z(a) > r(\alpha).$$

Informally, $def'(t_X)$ is greater than or equal to α in c' if and only if every assignment (t_X, t_Z) to (X, Z), with possibility $\pi_Z(t_Z)$ strictly greater than $1 - \alpha$, has preference $def(t_X, t_Z)$ greater than or equal to α in c. Thus, the higher is $def'(t_X)$, the more assignments to uncontrollable variables in c produce a preference that is higher than a given threshold.

Notice that def' in the formula above is computed by applying the *max* operator between preferences and possibilities. This can be done, assuming *commensurability* between preferences and possibilities, since the scale of the preferences and the scale of the possibilities are equal and since both fuzzy preferences and possibilities deal with vague knowledge.

If the uncontrollable variables $z_1, \ldots, z_k \in Z$ are logically independent from each other, the knowledge about each z_j is completely described by the possibility distributions π_{z_j}, and the joint possibility π_Z is as follows: $\forall a = (a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$, $\pi_Z(a) = min_{\{j=1,\ldots,k\}} \pi_{z_j}(a_j)$.

Example 2 Consider the constraint $c = \langle f, \{x, y, z\}\rangle$ in Figure 2 (a). The robustness constraint obtained from it is the constraint $\langle fr, \{x, y\}\rangle$ shown in Figure 2 (b). Consider for example the preference value associated to the assignment $(x = 1, y = 2)$, i.e., $fr(x = 1, y = 2) = 0.6$. By Property PO, for all the values t_i of z with possibility $\pi_Z(t_i) > 1 - 0.6 = 0.4$, i.e., in this case for the value $z = 4$, we have $f(x = 1, z = t_i) \geq 0.6$ in c. □

5.2 Projection constraints

Projection constraints are added to the problem in order to encode part of the information contained in the constraints in C_{cu}. In particular, as we will show in Example 5, they are useful to guarantee that the preference degree of a solution, say $pref(s)$, is a value that could be obtained in the given UFCSP.

The set of projection constraints C_{proj} is defined by function $ComputeProjectionConstraint$ in algorithm *SP*. Such a function takes in input a fuzzy constraint $c = \langle def, con\rangle$ in C_{cu}, such that $con \cap V_c = X$ and $con \cap V_u = Z$, and it returns a fuzzy constraint $\langle defp, X\rangle$, where $defp(t_X) = max_{\{t_Z \in A_Z | \pi_Z(t_Z) > 0\}} def(t_X, t_Z)$. In other words, $defp(t_X)$ is the best preference that could be reached in c for the assignment t_X to X, when we consider the various assignments t_Z to Z taken from the domain A_Z of Z.

Notice that this value has the form of the optimistic possibilistic expected utility (Dubois & Prade, 1995; Dubois et al., 1998, 1996), that is, $defp(t_X) = max_{\{t_Z \in A_Z} min(def(t_X, t_Z), \pi_Z(t_Z))$, where we consider only assignments t_Z of the uncontrollable variables with $\pi_Z(t_Z) > 0$ and we assume that all of them have a possibility degree of 1.

Example 3 Consider the constraint $c = \langle f, \{x, y, z\}\rangle$ in Figure 2 (a). The projection constraint obtained from it is the constraint $\langle fp, \{x, y\}\rangle$ shown in Figure 2 (b). Let us consider for example the value $fp(x = 1, y = 1) = 0.5$. It recalls us that in c the assignment $(x = 1, y = 1)$ has at most preference 0.5. □

5.3 An example

We will now illustrate with an example how algorithm *SP* works.

Example 4 Let us consider the UFCSP, say Q, in Figure 2 (a). Figure 2 (b) shows the corresponding problem Q' obtained by *SP* from Q. Notice that Q' is defined only on the controllable variables of Q, namely x, y and w. The set of constraints C_c contains the constraint $\langle q, \{x, w\}\rangle$. The set of constraints C_{proj} contains the constraint $\langle fp, \{x, y\}\rangle$, obtained projecting constraint $\langle f, \{x, y, z\}\rangle$ of Q on the controllable variables x and y. The set of constraints C_{rob}, instead, contains the constraint $\langle fr, \{x, y\}\rangle$. □

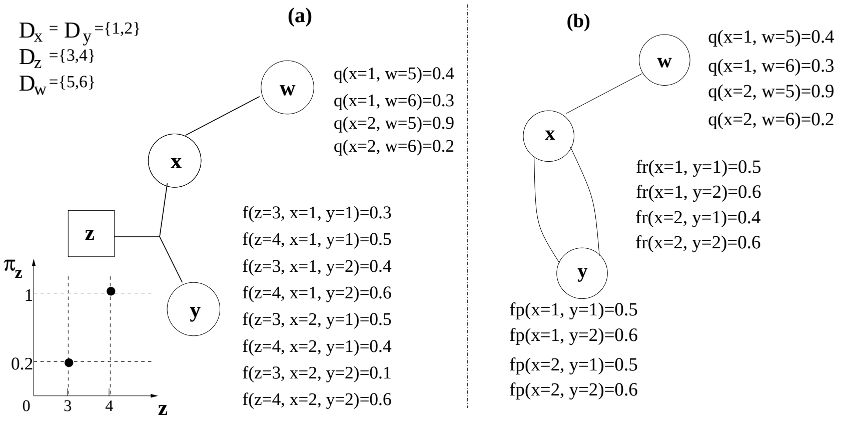


Figure 2: How *SP* works.

6. Preference and robustness

We are now ready to define the preference and the robustness of a solution in an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset\rangle$. The main idea is to use algorithm *SP* to produce the RFCSP $Q' = \langle S_{FCSCP}, V_c, C^*, C_{rob}\rangle$, where $C^* = C_c \cup C_{proj}$, and then to associate to each solution of Q' a pair composed by a degree of preference and a degree of robustness.

Definition 15 (preference) Given a solution s of an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset\rangle$, let $Q' = \langle S_{FCSP}, V_c, C_c^*, C_{rob}\rangle$, where $C_c^* = C_c \cup C_{proj}$, the RFCSP obtained from Q by algorithm *SP*. Then the preference of s is

$$pref(s) = min_{\{\langle def, con\rangle \in C_c^*\}} def(s \downarrow_{con}).$$

□

In words, the preference of a solution is the preference obtained by combining all the preferences of the subtuples of the solution over the constraints in $C_c \cup C_{proj}$ of the problem Q' returned by algorithm *SP*. Thus, the preference of a solution summarizes all the preferences in the controllable part (since it considers constraints in C_c), and it gives a value that can be obtained for at least one way in which Nature will assign values to uncontrollable variables (since it considers constraints in C_{proj}).

In the following we will sometimes need to use a preference value that we call *projection preference*. More precisely, we will denote the projection preference of a solution s with $proj(s) = min_{\{\langle def,con\rangle \in C_{proj}\}}\ def(s \downarrow_{con})$.

Considering also projection constraints in the computation of a solution's preference is important to avoid having solutions s with preference better than the best preference that can result from the C_{cu} constraints. In fact, it is not desirable to associate to a solution a preference that can be never obtained from the C_{cu} constraints.

Example 5 Consider the UFCSP Q shown in Figure 2 (a) and the solution $s = (x = 2, y = 1, z = 5)$ in Figure 3 (b). If we don't consider the projection constraint $\langle fp, \{x, y\}\rangle$ in the computation of the preference degree of s, then we will associate to s a preference value 0.9, that is strictly higher than the best preference value that can be obtained for s in Q, i.e., 0.5. Instead, if we add this projection constraint, we associate to s a preference value $0.5 = min(0.9, 0.5)$, that can be obtained in Q. □

Definition 16 (robustness) Given a solution s of an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset\rangle$, let $Q' = \langle S_{FCSP}, V_c, C_c^*, C_{rob}\rangle$, where $C_c^* = C_c \cup C_{proj}$, the RFCSP obtained from Q by algorithm *SP*. Then the robustness of s is

$$rob(s) = min_{\{\langle def,con\rangle \in C_{rob}\}} def(s \downarrow_{con}).$$

□

In words, the robustness of a solution is the preference obtained by combining all the preferences of the subtuples of the solution over the constraints in C_{rob} of the problem Q'.

The notion of robustness given in Definition 16 satisfies all the properties that we have stated as desirable in Section 4. In particular, given a solution s, the value $rob(s)$, as required in Section 4, depends, by construction of C_{rob}, on the preferences in the constraints in C_{cu} relating controllable and uncontrollable variables to s, and on the possibility distributions of such uncontrollable variables.

Moreover, as required in Section 4, $rob(s)$ generalizes the intuitive measure of robustness, which considers the worst (i.e., in the fuzzy case, the minimal) preference which can be obtained by s on the constraints involving controllable and uncontrollable variables. In fact, if there is no additional information on the uncontrollable variables in Z, we are in the case of complete ignorance, then the possibilities of every assignment t_Z to Z are equal to 1 (i.e., $\pi_Z(t_Z) = 1$, $\forall t_Z$), thus $rob(s) = min_{\{<def',con'> \in C_{rob}\}} def'(s \downarrow_{con'})$, where $def'(s \downarrow_{con'}) = min_{t_Z \in A_Z} max(def(s \downarrow_{con'}, t_Z), r(\pi_Z(t_Z))) = min_{t_Z \in A_Z} def(s \downarrow_{con'}, t_Z)$, since $r(\pi_Z(t_Z)) = r(1) = 1 - 1 = 0$.

Also, it satisfies properties 1 and 2 presented in Section 4.

Proposition 1 *The definition of robustness given in Definition 16 satisfies properties 1 and 2.*

Proof: In the Appendix.

The notion of robustness given in Definition 16 generalizes, as shown above, the intuitive notion of robustness which considers the worst preference which can be obtained by s on the constraints involving controllable and uncontrollable variables, while taking into account also the possibility distributions associated to the uncontrollable variables. Neglecting such additional information can

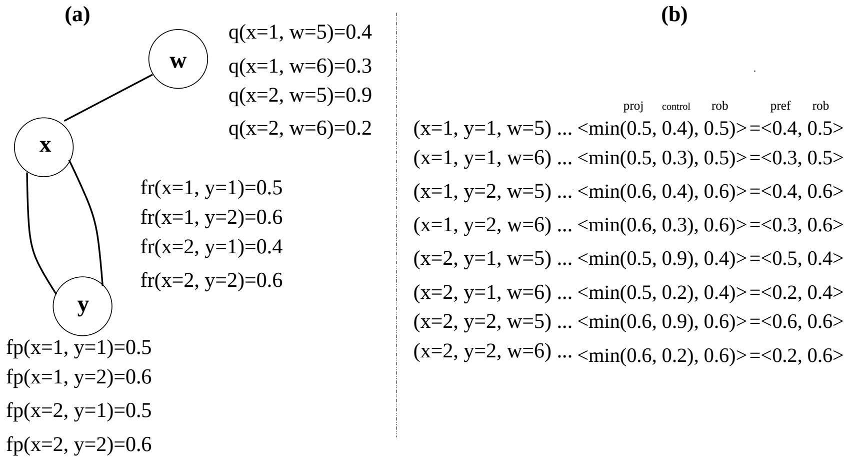

(a)
w
x
y
q(x=1, w=5)=0.4
q(x=1, w=6)=0.3
q(x=2, w=5)=0.9
q(x=2, w=6)=0.2
fr(x=1, y=1)=0.5
fr(x=1, y=2)=0.6
fr(x=2, y=1)=0.4
fr(x=2, y=2)=0.6
fp(x=1, y=1)=0.5
fp(x=1, y=2)=0.6
fp(x=2, y=1)=0.5
fp(x=2, y=2)=0.6
(b)
proj control rob pref rob
(x=1, y=1, w=5) ... <min(0.5, 0.4), 0.5>=<0.4, 0.5>
(x=1, y=1, w=6) ... <min(0.5, 0.3), 0.5>=<0.3, 0.5>
(x=1, y=2, w=5) ... <min(0.6, 0.4), 0.6>=<0.4, 0.6>
(x=1, y=2, w=6) ... <min(0.6, 0.3), 0.6>=<0.3, 0.6>
(x=2, y=1, w=5) ... <min(0.5, 0.9), 0.4>=<0.5, 0.4>
(x=2, y=1, w=6) ... <min(0.5, 0.2), 0.4>=<0.2, 0.4>
(x=2, y=2, w=5) ... <min(0.6, 0.9), 0.6>=<0.6, 0.6>
(x=2, y=2, w=6) ... <min(0.6, 0.2), 0.6>=<0.2, 0.6>

Figure 3: Solutions of an UFCSP with their preference and robustness degrees.

lead to unreasonable judgments, as illustrated by Example 6. In particular, using the intuitive definition of robustness we could consider as bad a situation that behaves well in almost all cases, and that is bad only in one very unlikely case.

Example 6 Consider an UFCSP composed only by the constraint $\langle f, \{x, z\}\rangle$ linking the controllable variable x with the uncontrollable variable z. Assume that the preference function f is defined on (x, z) as follows: $f(d, a_1) = 0.9$, $f(d, a_2) = 0.9$, $f(d, a_3) = 0.9$, $f(d, a_4) = 0.2$. Assume also that assignments a_1, a_2 and a_3 to variable z have possibility equal to 1, and that assignment a_4 has possibility 0.1. If we consider the notion of robustness mentioned before, that doesn't consider the possibilities, then we give the solution $x = d$ a low robustness value $0.2 = min(0.9, 0.9, 0.9, 0.2)$, even if it behaves badly only in one unlikely case. Instead, considering our notion, for solution $x = d$ we have a higher value of robustness, i.e., $0.9 = min_{\{a_1,a_2,a_3,a_4\}}(max(0.9,\, 0)$, $max(0.9,$ $0), max(0.9,\, 0)$, $max(0.2,\, 0.9))$, which is coherent with the fact that solution $x = d$ behaves well in most possible cases. □

We will now show via an example how to compute the preference and the robustness values of the solutions of an UFCSP.

Example 7 Let us consider the UFCSP shown in Figure 2 (a). Figure 3 (b) shows all the solutions of Q with their preference and robustness values. □

7. Semantics

In this section we propose various semantics for ordering the solutions with respect to their preference and robustness values, and we check if these semantics, as well as the semantics defined in (Dubois et al., 1996), satisfy the desired properties on solution ordering.

Once each solution s is associated with two values, the preference degree $pref(s)$ and the robustness degree $rob(s)$, there can be several ways to order the solutions. We will now propose various approaches, which differ on their attitude toward risk.

Definition 17 (semantics) Given an UFCSP Q, consider a solution s with preference degree $pref(s)$ and robustness degree $rob(s)$. Each semantics associates to s the ordered pair $\langle a_s, b_s \rangle$ as follows:

- **Risky** (R), **Diplomatic** (D): $\langle a_s, b_s \rangle$=$\langle pref(s), rob(s) \rangle$;
- **Safe** (S): $\langle a_s, b_s \rangle$=$\langle rob(s), pref(s) \rangle$;
- **Risky-Mix** (R1): $\langle a_s, b_s \rangle$=$\langle min(pref(s), rob(s)), pref(s) \rangle$;
- **Safe-Mix** (S1): $\langle a_s, b_s \rangle$=$\langle min(pref(s), rob(s)), rob(s) \rangle$.

Given two solutions s and s', let $\langle a_s, b_s \rangle$ and $\langle a_{s'}, b_{s'} \rangle$ represent the pairs associated to the solutions by a semantics. The Risky, Safe, Risky-Mix, and Safe-Mix semantics work as follows:

- if $a_1 > a_2$ then $\langle a_1, b_1 \rangle >_J \langle a_2, b_2 \rangle$ (and the opposite for $a_2 > a_1$)
- if $a_1 = a_2$ then
  - if $b_1 > b_2$ then $\langle a_1, b_1 \rangle >_J \langle a_2, b_2 \rangle$ (and the opposite for $b_2 > b_1$)
  - if $b_1 = b_2$ then $\langle a_1, b_1 \rangle = \langle a_2, b_2 \rangle$;

where $J \in \{R, S, R1, S1\}$.

For the Diplomatic semantics, we have a different behavior:

- if $a_1 \leq a_2$ and $b_1 \leq b_2$ then $\langle a_1, b_1 \rangle \leq_D \langle a_2, b_2 \rangle$ (and the opposite for $a_2 \leq a_1$ and $b_2 \leq b_1$);
- if $a_1 = a_2$ and $b_1 = b_2$ then $\langle a_1, b_1 \rangle = \langle a_2, b_2 \rangle$;
- else $\langle a_1, b_1 \rangle \bowtie \langle a_2, b_2 \rangle$, where $\bowtie$ means incomparable. □

As it can be seen by Definition 17, all semantics, except Diplomatic, can be regarded as a lexicographic ordering on pairs $\langle a_s, b_s \rangle$ with the first component as the most important feature. Diplomatic, instead, is a Pareto ordering on the pairs.

Let us describe more in detail such semantics:

- Risky: this semantics considers $pref(s)$ as the most important feature. Informally, the idea is to give more relevance to the preference degree that can be reached in the best case of the uncertain part, considering less important a high risk of being inconsistent. Hence we are risky, since we disregard almost completely the uncertain part of the problem. This could be useful to reason with uncertain problems where the uncontrollable part is almost irrelevant w.r.t. the controllable one. In this case we are interested into finding solutions that guarantee an high level of preference in the controllable part even if such a preference could be worsened in some scenario of the uncontrollable part. Only when we have two or more solutions with same best preference degree we consider the robustness degree, and so the uncontrollable part of the problem, to choose one among these solutions.

- Safe: this semantics represents the opposite attitude with respect to the previous one, since it considers $rob(s)$ as the most important feature. Informally, the idea is to give more importance to the robustness level that can be reached, considering less important having a high preference. In particular, in this case we consider a solution better than another one if its robustness is higher, i.e., if it guarantees an higher number of scenarios with an higher preference. This semantics considers the preference degree of a solution only for ordering solutions having the same robustness. This can be useful for reasoning with uncertain problems when we are mainly interested in the part of the problem that we cannot control. In this case we want to find the most robust solution independently from its preference degree. If the chosen solution has a very bad preference degree, we could modify that solution, if we want, since we can decide the controllable part.

- Risky-Mix: this semantics tries to overcome the myopic attitude of Risky, which concentrates only on the preference degree (except when there is a tie), by first considering the ordering generated by the minimum of $pref(s)$ and $rob(s)$. This allows to avoid considering as good, solutions which will give a low overall preference in most of the possible scenarios.

- Safe-Mix: this semantics mitigates the relevance given to robustness in Safe. In fact, not considering the minimum value between $pref$ and rob before focusing on rob, as the Safe semantics does, can lead to consider, as optimal, solutions which have a poor overall preference despite guaranteeing high preference on the constraints involving uncontrollable variables.

- Diplomatic: this semantics aims at giving the same importance to the two aspects of a solution: preference degree and robustness. As mentioned above, the Pareto ordering on pairs $\langle a_s, b_s \rangle$ is adopted. The idea is that a pair is to be preferred to another only if it wins both on preference and robustness, leaving incomparable all the pairs that have one component higher and the other lower. Contrarily to the Diplomatic, the other semantics produce a total order over the solutions.

All semantics differ with respect to the attitude toward risk they implement and with respect to the point of view from which they consider the problem.

In particular, in the Risky and Safe semantics, for each solution, the two aspects, i.e., how well the solution performs respectively on the controllable part of the problem and on the constraints involving uncontrollable variables, are kept separated. The relation between how a solution satisfies the two aspects is ignored. This can be seen as a myopic attitude which focuses on the preference of the controllable part in Risky, and on the compatibility with uncertainty in Safe. It should be noticed, however that, given a solution s, the preference obtained by s on the constraints involving only controllable variables has absolutely no impact on $rob(s)$, while the compatibility of the solution with uncertain events is taken into account (through the projection constraints) for $pref(s)$. In this sense, Risky can be regarded as less myopic than Safe.

A more global view of the problem characterizes instead the Risky-Mix, Safe-Mix and Diplomatic semantics. In fact, in the first two, considering the minimum of $pref(s)$ and $rob(s)$ allows one to order the solutions first with respect to their predominant aspect, that is the one on which they have a worst performance. Diplomatic, instead, considers both aspects separate but with the same importance.

Example 8 Figure 3 (b) shows a solution of the FCSP in Figure 3 (a) which is optimal according to all the semantics described in Definition 17. □

Let us now consider an example that explains the differences between the various semantics.

Mixed	Risky	Safe	Dipl.	Risky-Mix	Safe-Mix
$=$	$<,>, =$	$<,>,=$	$<,>,=, \bowtie$	$<,>, =$	$<,>, =$
$>$	$<,>$	$<,>$	$>, \bowtie$	$>$	$>$

Table 1: The solution ordering produced by the Mixed semantics compared to that of Risky, Safe, Diplomatic, Risky-Mix and Safe-Mix semantics.

Example 9 Let us consider two solutions of an UFCSP, s_1 and s_2, such that $pref(s_1) = 0.3$, $rob(s_1) = 0.5$, $pref(s_2) = 0.5$ and $rob(s_2) = 0.3$. According to the semantics defined above we have the following orderings: $s_1 <_{R,R1} s_2$, $s_1 >_{S,S1} s_2$, and $s_1 \bowtie_D s_2$.

If we consider two solutions, s_3 and s_4, such that $pref(s_3) = 0.5$, $rob(s_3) = 0.3$, $pref(s_4) = 0.6$ and $rob(s_4) = 0.2$, then $s_3 <_R s_4$, $s_3 >_{S,S1,R1} s_4$, and $s_3 \bowtie_D s_4$. □

Another semantics, that we will consider in the following, is the semantics adopted in (Dubois et al., 1996) to order the solutions of UFCSPs. Such a semantics, that we call **Mixed**, is such that, given $A1 = (pref_1, rob_1)$ and $A2 = (pref_2, rob_2)$, $A1 \succ_{Mixed} A2$ iff $min(pref_1, rob_1) > min(pref_2, rob_2)$. In words, according to this semantics, a solution s_1 with preference $pref_1$ and robustness rob_1, is better than a solution s_2 with preference $pref_2$ and robustness rob_2 iff the minimum between $pref_1$ and rob_1 is greater than the minimum between $pref_2$ and rob_1.

7.1 Comparing the semantics

In this section we will compare the semantics we have considered in terms of the ordering they produce over the solutions.

Table 1 shows how a pair of solutions, which is ordered in a given way by Mixed, is ordered by the other semantics. The first row of Table 1 indicates that if a pair of solutions is equally preferred by Mixed, it can be equally preferred or ordered in any way for Risky, Safe, Risky-Mix and Safe-Mix, and it can also be incomparable for Diplomatic.

Example 10 Consider two solutions, s_1 and s_2 such that $pref(s_1) = 0.5$, $rob(s_1) = 0.7$, $pref(s_2) = 0.7$, and $rob(s_2) = 0.5$, then we have $s_1 =_{Mixed} s_2$, $s_1 <_{R,R1} s_2$, $s_1 >_{S,S1} s_2$, and $s_1 \bowtie s_2$. Consider, instead, two solutions s_3 and s_4 such that $pref(s_3) = 0.2$, $rob(s_3) = 0.2$, $pref(s_4) = 0.5$, and $rob(s_4) = 0.2$, then $s_4 =_{Mixed,S1} s_3$, while $s_4 >_{R,R1,S,D} s_3$. □

The second row of Table 1 states that, if a pair is ordered in some way by Mixed, then it can be ordered in the same way by all the semantics, or in the opposite way in Risky and Safe, or it can be incomparable in Diplomatic.

Example 11 Consider six solutions, s_1, s_2, s_3, s_4, s_5 and s_6 such that: $pref(s_1) = 0.4$, $rob(s_1) = 0.3$, $pref(s_2) = 0.5$, $rob(s_2) = 0.4$, $pref(s_3) = 0.5$, $rob(s_3) = 0.3$, $pref(s_4) = 0.4$, $rob(s_4) = 0.4$, $pref(s_5) = 0.3$, $rob(s_5) = 0.8$, $pref(s_6) = 0.4$, and $rob(s_6) = 0.7$. Then we have $s_1 <_{Mixed,R,S,D,R1,S1} s_2$, $s_3 <_{Mixed} s_4$, $s_3 >_R s_4$, $s_5 <_{Mixed} s_6$ and $s_5 >_S s_6$. □

Notice that two solutions, which are strictly ordered in Mixed, cannot be equally preferred with respect to Safe or Risky semantics. In fact, two solutions are equally preferred for Safe and Risky (and Diplomatic) only if they have the same preference and the same robustness, and thus the same minimum.

A pair ordered by Mixed can either maintain its ordering or become incomparable according to Diplomatic (as shown in Table 1). Moreover, Risky-Mix and Safe-Mix refine the ordering given by Mixed, i.e., they can order tuples that are considered equal for Mixed, but they never reverse the Mixed ordering. All these results are shown in the following proposition.

Proposition 2 *Consider two solutions of an UFCSP, say s_1 and s_2. If $s_1 >_{Mixed} s_2$, then either $s_1 >_D s_2$ or $s_1 \bowtie_D s_2$. Also, $s_1 >_{R1,S1} s_2$.*

Proof: In the Appendix.

Notice that from the second part of Proposition 2 it derives that the set of optimal solutions according to Mixed is a superset of the set of optimal solutions of Risky-Mix and Safe-Mix.

7.2 Desired properties on the solution ordering

We now check if the preferential orderings over the solutions of an UFCSP induced by the semantics presented in Definition 17 satisfy Properties 3, 4 and 5 of Section 4.

Property 3 is not satisfied by the preferential orderings induced by the various semantics. However, it is possible to show that, if we combine all the constraints in C_{cu} involving both controllable and uncontrollable variables, and if compute, starting from this constraint, the projection constraint and the robustness constraint, the preferential ordering induced by Risky and Risky-Mix semantics satisfy such a property.

In our solver, that we will describe in detail in Section 8, we have decided to avoid to perform the combination of all the constraints in C_{cu}, since it can be very expensive, and thus we have computed a projection constraint and a robustness constraint for every constraint in C_{cu}.

Proposition 3 *The preferential orderings over the solutions of an UFCSP induced by Risky, Risky-Mix, Safe, Safe-Mix and Diplomatic do not satisfy Property 3.*

Proof: In the Appendix.

Proposition 4 *Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset \rangle$. If we compute the constraint obtained by combining all the constraints in C_{cu} and, starting directly from such a constraint, we compute the projection constraint over the controllable variables of C_{cu} and the robustness constraint, then the preferential orderings over the solutions of Q induced by Risky and Risky-Mix satisfy Property 3.*

Proof: In the Appendix.

Proposition 5 *The preferential orderings over the solutions of an UFCSP induced by Risky, Risky-Mix, Safe and Diplomatic satisfy Property 4, while the one induced by Safe-Mix does not satisfy Property 4.*

Proof: In the Appendix.

Proposition 6 *The preferential orderings over the solutions of an UFCSP induced by Risky, Safe, Safe-Mix and Diplomatic satisfy Property 5, while the one induced by Risky-Mix does not satisfy Property 5.*

Proof: In the Appendix.

Summarizing, the preferential ordering over the solutions of an UFCSP induced by Risky satisfies Properties 4 and 5, the one induced by Risky-Mix satisfies Properties 4, those induced by Safe and Diplomatic satisfy 4 and 5, and the one induced by Safe-Mix satisfies only Property 5. Moreover, if we perform the combination of all the constraints of C_{cu} before computing the robustness and the projection constraints, then the preferential orderings induced by Risky and Risky-Mix satisfy also Property 3.

Although Properties 3, 4 and 5 are desirable, there are semantics that does not any satisfy them. It is possible to show that Mixed semantics does not satisfy Properties 3, 4 and 5. Before showing this, we give a result which will be useful in the proof of the following proposition.

Theorem 1 *For every solution s of an UFCSP, $rob(s) \leq proj(s)$.*

Proof: In the Appendix.

Proposition 7 *The preferential ordering over the solutions of an UFCSP induced by the Mixed semantics does not satisfy Properties 3, 4 and 5.*

Proof: In the Appendix.

This behavior depends on the use of the *min* operator, since it forgets about all the elements which are higher than the minimum. This is usually called the ”drowning effect” (Dubois & Prade, 1993).

	Mixed	Risky	Safe	Dipl.	Safe-Mix	Risky-Mix
Property 1	X	X	X	X	X	X
Property 2	X	X	X	X	X	X
Property 3		X^*				X^*
Property 4		X	X	X		X
Property 5		X	X	X	X	

Table 2: Properties satisfied in the various semantics. The presence of X in a cell (Property i, S) denotes that semantics S satisfies Property i. The presence of X^* in a cell (Property i, S) denotes that, if we compute the constraint obtained by combining all the constraints in C_{cu} and, starting from such a constraint, we compute the projection constraint over the controllable variables of C_{cu} and the robustness constraint, semantics S satisfies Property i.

Table 2 summarizes which properties hold in the various semantics. By looking at Table 2 we can make the following remarks.

- Considering the first two rows, we see that the properties pertaining the definition of robustness are satisfied by all the semantics. Indeed all semantics use the same value, $rob(s)$, as a measure of robustness. We recall that Property 1 states that an increase in the preferences on constraints involving uncontrollable variables results in an increase of robustness, assuming that possibility distribution is kept fixed. The same result can be obtained by lowering the possibilities while maintaining the preferences fixed (property 2). Proposition 1 shows that the definition of rob, given in Definition 16, satisfies these properties.

- Property 3 measures the coherence of the ordering produced by the semantics with the original one defined on the UFCSP. Notice that not necessarily the absence of such coherence should be considered as a drawback. In fact, while, at first sight, it may seem desirable to prefer an assignment to controllable variables which outperforms another one in every circumstance, this is not so obvious when the performance measure is the min of the preference over all the constraints. In particular, it may be reasonable to sacrifice this property in order to allow a higher discriminating power among the two fundamental aspects of the solutions, which are the preference and the robustness with respect to uncertainty.

- The last two properties are satisfied when, given two solutions that have the same robustness then their ordering is determined by the preference degree (Property 4) and, given two solutions that have the same preference then their ordering is determined by the robustness (Property 5). Both of these properties are satisfied by Risky, Safe and Diplomatic semantics, since such semantics consider the two features separately and independently. The other two semantics, Risky-Mix and Safe-Mix, which first consider the min of the two values, allow to discriminate only with respect to the feature which appears as the second element of the pair. This can be explained in terms of a trade off between the discrimination power of the semantics and its coherence with the original ordering of the UFCSP.

8. A solver for UFCSPs

We now present a solver for UFCSPs. It takes in input an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, C_u \rangle$, and it returns an optimal solution according to one of the semantics defined in Section 7. Note that, for simplicity, we consider only binary constraints in C_u. However, we don't lose generality, since we can always reformulate a problem with non-binary soft constraints in a problem with only binary constraints (Rossi et al., 2006).

This solver consists of the following phases:

1. *Preprocessing:* the goal of this phase is to connect every uncontrollable variable, that is not directly connected to a controllable one, to some controllable variable. This is achieved by adding constraints which are induced by existing constraints in C_u and C_{cu}. This is needed to avoid loss of information, since algorithm *SP* removes all the uncontrollable variables from the problem only considering the information contained in constraints between controllable and uncontrollable variables. We indicate with C'_{cu} the set of constraints including those in C_{cu} and the induced ones.

2. *Enforcing generalized soft arc consistency (GSAC) on $C'_{cu} \cup C_u$:* this step mitigates the loss of information contained in the constraints in C_u, that will be removed by Algorithm *SP*, and the loss of information about the correlations among constraints in C'_{cu}, since they will be

processed separately by *SP* even if they share some variable. In fact, it allows to propagate the information of the constraints in $C'_{cu} \cup C_u$ over their variables. In practice, some preferences associated to the values in the domains of such variables may be lowered.

3. *SP algorithm* (presented in Section 5): this algorithm removes all the uncontrollable variables and, thus, all the constraints involving them (C'_{cu} and C_u), by adding new constraints C_{rob} and C_{proj}, i.e., the robustness and projection constraints, obtained from the constraints in C'_{cu}. We note that this may cause a loss of information. Algorithm *SP* returns the current fuzzy problem without the uncontrollable part and with the new constraints.

4. *Enforcing soft arc consistency (SAC) or its generalized version (GSAC) separately on C_c, C_{proj} and C_{rob}* thus generating three independent sets of unary constraints: this propagation step may reduce even further the search space. In particular, in the first case only binary constraints are propagated, while, in the second case, constraints of any arity are propagated. We consider separately constraints in C_c, C_{proj} and C_{rob} in order to follow the requirements of the various semantics.

5. *Branch and Bound search:* a branch and bound search procedure is used to find an optimal solution according to one of the semantics defined in Section 7.

8.1 Preprocessing

The preprocessing phase takes in input an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, C_u \rangle$, with $V_u = Z_c \cup Z_{nc}$, where Z_c is the set of uncontrollable variables, that are *connected* directly to a controllable variable, and Z_{nc} is the set of uncontrollable variables in C_u, that are *not connected* directly with any controllable variable. Then, it returns an UFCSP, that is like Q except that there are also new constraints, where all the uncontrollable variables in Z_{nc} are connected to the controllable part.

In particular, if C_u is empty, the algorithm simply returns the UFCSP given in input, since there are no constraints involving only uncontrollable variables. Otherwise, it returns the UFCSP $\langle S_{FCSP}, V_c, V_u, \pi, C_c, C'_{cu}, C_u \rangle$, with $C_u = \emptyset$, that has been obtained from the given one by adding induced constraints between every variable z_{nc} in Z_{nc} and at least one controllable variable in V_c. In particular, for every variable $z_{nc} \in Z_{nc}$ connected to a variable $z_c \in Z_c$ that is connected to a set X of controllable variables, we compute the induced constraint between z_{nc} and X, by combining the constraint between z_{nc} and z_c with the constraint between z_c and X, and by projecting the resulting constraint over the variables in $X \cup z_{nc}$.

Example 12 Consider the UFCSP Q shown in Figure 4. There is a variable $z_1 \in Z_c$, which is involved in a constraint $c_1 \in C_{cu}$. Let us call X the set $\{x_1, x_2\}$ of the controllable variables involved in c_1. Also, there is a variable $z_2 \in Z_{nc}$ connected to z_1 by a constraint $c_2 \in C_u$. Assume that x_2, z_1, and z_2 have domain $\{a, b\}$ and that x_1 has domain $\{c, d\}$. Assume also that $c_1 = \langle f_1, \{x_1, x_2, z_1\} \rangle$, where $f_1(c, a, a) = 0.4$, $f_1(c, a, b) = 0.3$, $f_1(c, b, a) = 0.2$, and $f_1(c, b, b) = 0.1$, $f_1(d, a, a) = 0.7$, $f_1(d, a, b) = 0.4$, $f_1(d, b, a) = 0.3$, and $f_1(d, b, b) = 0.2$, and that $c_2 = \langle f_2, \{z_1, z_2\}$, where $f_2(a, a) = 0.5$, $f_2(a, b) = 0.6$, $f_2(b, a) = 0.7$, and $f_2(b, b) = 0.2$. The preprocessing phase creates a new constraint, that we call c_3, between z_2 and X. The constraint c_3 is $\langle f_3, \{x_1, x_2, z_2\} \rangle$, where the preference of every assignment (a', b', c') to (x_1, x_2, z_2) is obtained

Algorithm 2: *Preprocessing*

```
input : $Q = \langle S_{FCSP}, V_c, Z_c \cup Z_{nc}, \pi, C_c, C_{cu}, C_u \rangle$: UFCSP
output: $Q' = \langle S_{FCSP}, V_c, Z_c \cup Z_{nc}, \pi, C_c, C'_{cu}, \emptyset \rangle$: UFCSP

$C'_{cu} \leftarrow C_{cu}$;
if $C_u \neq \emptyset$ then
    foreach $z_{nc} \in Z_{nc}$ do
        $z_c \leftarrow$ a variable of $V_c$ connected to $z_{nc}$;
        $X \leftarrow$ set of controllable variables of a constraint involving $z_c$;
        $C'_{cu} \leftarrow C'_{cu} \cup Generate\_Induced\_Constraints(z_{nc}, X)$;
return $Q' = \langle S_{FCSP}, V_c, Z_c \cup Z_{nc}, C'_{cu}, \emptyset \rangle$;
```

by computing $max_{\{a \in D\}} min(f_1(a', b', a), f_2(a, c'))$, where D is the domain of z_1. Hence, for example, the preference of the tuple (c, a, a) to (x_1, x_2, z_2) is $f_3(c, a, a) = max(min(f_1(c, a, a), f_2(a, a)), min(f_1(c, a, b), f_2(b, a)) = max(min(0.4, 0.5), min(0.3, 0.7)) = max(0.4, 0.3) = 0.4$. □

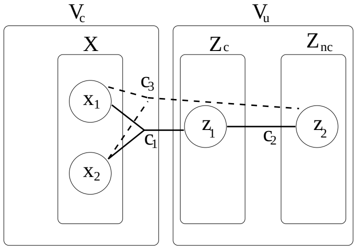


Figure 4: Preprocessing.

8.2 SAC and GSAC

In Algorithm 3 we show Algorithm *GSAC*, that performs a generalized soft arc-consistency (Rossi et al., 2006) over a set of constraints. It takes in input a set S of constraints, and at the end S is modified in a way such that it is generalized arc consistent. At every step of the **while** loop, a constraint c is extracted from the set S by the procedure $ExtractConstraint$, and for every variable x_i of the constraint c the variable denoted $modified$ is initialized to the value $false$. Such a variable indicates if the preference of a value a of the variable x_i has been changed. Then, it considers every value a in the domain D_i of x_i, computes all the tuples which extend the current value a of x_i via the procedure $Extensions$, and it projects such tuples on x_i via the procedure $Project$, thus obtaining a new preference value. If this value is different from $f_i(a)$, that is, the preference of such value a of x_i in the constraint $\langle f_i, x_i \rangle$ of S, then the algorithm inserts in S, via the procedure $ObtainConstraints$, all the constraints which involve the variable x_i, since they should be reconsidered. The **while** loop is repeated until S is empty, that is, until all the constraints in S have been considered, and there is no other change in the preference values associated to the values of the variables. In this case S is generalized arc-consistent.

If we want to propagate only binary constraints, we can use Algorithm 3, requiring that the set S of the considered constraints is composed only by binary constraints, and that the operations $ExtractConstraint$ and $ObtainConstraints$ consider only binary constraints. In this case we denote Algorithm 3 with *SAC*.

Algorithm 3: *GSAC* (resp., *SAC*)

```
Input: S: set of constraints (resp., set of binary constraints)
while $S \neq \emptyset$ do
    c ← $ExtractConstraint(S)$;
    foreach $x_i \in c$ do
        modified ← false;
        foreach $a \in D_i$ do
            $T \leftarrow Extensions(a, c)$;
            $v^* \leftarrow Project(T, x_i)$;
            if $v^* \neq f_i(a)$ then
                $f_i(a) \leftarrow v^*$;
                $modified \leftarrow true$;
        if $modified = true$ then
            $S = S \cup ObtainConstraints(x_i) - \{c\}$;
```

Example 13 Let us consider an FCSP with three variables, x, y and z, with domain $\{a, b\}$. Assume that the set of constraints is composed by three constraints: $cx = \langle f_x, \{x\} \rangle$, where $f_x(a) = 0.9$, $f_x(b) = 0.1$, $cy = \langle f_y, \{y\} \rangle$, where $f_y(a) = 0.9$, $f_y(a) = 0.5$, and $cxy = \langle f_{xy}, \{x, y\} \rangle$, where $f_{xy}(a, a) = 0.8$, $f_{xy}(a, b) = 0.2$, $f_{xy}(b, a) = 0$, and $f_{xy}(b, a) = 0$. Applying *SAC* to this set of constraints, we obtain these new lower preference values in cx and cy: $f_x(a) = 0.8$, $f_x(b) = 0$, $f_y(a) = 0.8$, and $f_y(a) = 0.2$. Consider the assignment x to a, its new preference is obtained by computing $max(min(f_x(a),\ f_{xy}(a, a)),\ min(f_x(a),\ f_{xy}(a, b)) = max(min(0.9,\ 0.8),\ min(0.9,\ 0.2) = 0.8$. □

8.3 Branch and Bound

We now describe the Branch and Bound (*BB*) algorithm that has been implemented in our solver. It takes in input an UFCSP and one of the semantics described in Section 7, and it returns an optimal solution of the given UFCSP according to the ordering induced by the chosen semantics.

It follows the scheme of the classical Branch and Bound algorithm (Dechter, 2003) used for fuzzy preferences. Whenever a solution is found, its preference, if higher that those found before, is kept as a lower bound, L, for the optimal preference. 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., if the preference of the best solution in the subtree below t is worse than or equal to the preference of the best solution found so far, then the subtree below t is pruned.

Our *BB* algorithm adapts the classical one to our framework, where every solution is not associated to a single preference value, but to two values, representing its preference and its robustness, and where there are several semantics to order the solutions according to these two values. In particular, it associates to every solution s an ordered pair of values, that is composed by the preference level $pref(s)$ and the robustness $rob(s)$, and it uses the various semantics for comparing solutions. Also the upper bound $ub(t)$ of a partial solution t is given by a pair of values, that we call $ubpref(t)$ and $ubrob(t)$, that are respectively overestimations of the best preference and the best robustness of a solution extending t.

Such overestimations can be computed by considering both instantiated and non-instantiated variables (*PastFutureUpperBound*), or only the instantiated ones (*PastUpperBound*). In the first case we have $ubpref(t) = min(pref_i(t), pref_{ni}(t))$, and $ubrob(t) = min(rob_i(t), rob_{ni}(t))$, while in the second case we have that $ubpref(t) = pref_i(t)$, and $ubrob(t) = rob_i(t)$, where $pref_i(t)$ and $rob_i(t)$ are respectively the preference level and the robustness value of the part of the problem instantiated by t, while $pref_{ni}(t)$ and $rob_{ni}(t)$ are overestimations respectively of the preference level and of the robustness value of the non-instantiated part of problem. In particular, the values $pref_{ni}(t)$ and $rob_{ni}(t)$ are obtained combining all the best preference values of the extensions of t in $C_c \cup C_{proj}$ and in C_{rob}. More formally, given a partial assignment t,

- let $c_i = \langle def_i, con_i \rangle$ be the generic constraint of $C_c \cup C_{proj}$ (resp., C_{rob}) involving only variables instantiated in t with $i = 1, \ldots, k$, then $pref_i(t) = min_{\{i=1,\ldots,k\}} \ def_i(t \downarrow_{con_i})$ (resp., $rob_i(t) = min_{\{i=1,\ldots,k\}} def_i(t \downarrow_{con_i})$);
- let $c_j = \langle def_i, con_i \rangle$ be the generic constraint of $C_c \cup C_{proj}$ (resp., C_{rob}) involving both variables instantiated in t and variables non-instantiated in t, with $i = 1, \ldots, k$, then, let t_m, for $m = 1, \ldots, l$, be an extension of $t \downarrow_{con_j}$, that assigns values also to the variables that are non-instantiated of c_j, then $pref_{ni}(t) = min_{\{i=1,\ldots,k\}} \ (\sum_{m=1}^{l} def_j(t_m))$ (resp., $rob_{ni}(t) = min_{\{i=1,\ldots,k\}}(\sum_{m=1}^{l} def_j(t_m))$.

In order to state in a compact way the conditions that allow to prune the subtree below a partial assignment t in the various semantics, we use the ordered 4-tuple $\langle a_i(t), a_{ni}(t), b_i(t), b_{ni}(t) \rangle$ to represent the values associated to t. This tuple is $\langle pref_i(t), pref_{ni}(t), \ rob_i(t), \ rob_{ni}(t) \rangle$ in Risky and Diplomatic, $\langle rob_i(t), rob_{ni}(t), pref_i(t), \ pref_{ni}(t) \rangle$ in Safe, $\langle min(pref_i(t), rob_i(t)), min(pref_{ni}(t), \ rob_{ni}(t)), \ pref_i(t), \ pref_{ni}(t) \rangle$ in Risky-Mix, and $\langle min(pref_i(t), \ rob_i(t)), \ min(pref_{ni}(t), \ rob_{ni}(t)), \ rob_i(t), rob_{ni}(t) \rangle$ in Safe-Mix.

More formally, assume that the best solution found so far is associated with the pair $\langle lb_a, lb_b \rangle$, and that the partial assignment t is associated with the tuple $\langle a_i(t), a_{ni}(t), b_i(t), b_{ni}(t) \rangle$. Then, in our *BB* algorithm, if we consider *PastFutureUpperBound*, we prune the subtree below t if one of the following conditions, that we call conditions of type α, holds:

1. $a_i(t) < lb_a$;
2. $a_i(t) \geq lb_a$ and $a_{ni}(t) < lb_a$;
3. $a_i(t) \geq lb_a$, $a_{ni}(t) = lb_a$, and $b_i(t) < lb_b$;
4. $a_i(t) \geq lb_a$, $a_{ni}(t) = lb_a$, $b_i(t) = lb_b$, and $b_{ni}(t) \leq lb_b$.

Theorem 2 *When BB considers a semantics among Risky, Safe, Risky-Mix or Safe-Mix, overestimations via PastFutureUpperBound, and the pruning conditions of type α, it returns solutions that are optimal according to the chosen semantics.*

The solutions that are optimal both for Risky and Safe are optimal also for Diplomatic. However, some solutions, that are optimal for Diplomatic, may be not optimal for Risky and Safe. For example, if we have only three solutions s, s' and s'' resp., with values $(0.5, 0.8)$, $(0.6, 0.6)$, and $(0.8, 0.5)$, then s is optimal for Safe, s'' is optimal for Risky and both s, s' and s'' are optimal for Diplomatic. Hence there is a solution, i.e., s', that is neither Risky nor Safe optimal. We want to be able to find this kind of solutions. To compute such optimal solutions for Diplomatic, we need different pruning conditions, that are tighter than the previous ones. In fact, they require that both conditions on the first component and conditions on the second one are satisfied. In particular, if the current best solution is associated to a pair, we want to be able to replace such a solution also by a solution that is better only in one component, i.e., by a solution that may be incomparable with it. For example, if the current best is associated to the pair $(0.5, 0.8)$, then we want to be able to replace this solution with a new current best solution that is $(0.7, 0.3)$.

We recall that in Diplomatic, given a partial assignment t, $a_i(t) = pref_i(t)$, $a_{ni}(t) = pref_{ni}(t)$, $b_i(t) = rob_i(t)$, and $b_{ni}(t) = rob_{ni}(t)$. Optimal solutions can be obtained by pruning the subtree below t if one of the following conditions, that we call of type β, holds:

1. $a_i(t) \leq lb_a$ and $b_i(t) \leq lb_b$;
2. $a_i(t) \leq lb_a$, $b_i(t) \geq lb_b$, and $b_{ni}(t) \leq lb_b$;
3. $a_i(t) \geq lb_a$, $a_{ni}(t) \leq lb_a$, and $b_i(t) \leq lb_b$;
4. $a_i(t) \geq lb_a$, $a_{ni}(t) \leq lb_a$, $b_i(t) \geq lb_b$, and $b_{ni}(t) \leq lb_b$.

Theorem 3 *When BB considers the Diplomatic semantics, overestimations via PastFutureUpperBound, and the pruning conditions of type β, it returns solutions that are optimal according to Diplomatic.*

If instead we consider the heuristics *PastUpperBound* to compute overestimations, then the pruning conditions are similar to the ones considered before, but they don't consider the conditions with $a_{ni}(t)$ and $b_{ni}(t)$.

In our solver, we have implemented four versions of *BB*: two of them for Diplomatic and two of them for the other semantics, which exploit the different pruning conditions described before. The two versions for each semantics differ for the procedure used to compute the upper bounds: a version considers both instantiated and non-instantiated variables (*PastFutureUpperBound*), while the other one considers only the instantiated ones (*PastUpperBound*). In all the implemented *BB* versions, the next variable to be instantiated is the most constrained one, while the values in the variable domains are ordered in a descending preference order.

All the *BB* algorithms are composed by two main parts. There is a part where we test if the assignment t is a complete assignment. If this holds, then t is a solution of the problem, thus we have to check if it is better than the best solution obtained so far. If this holds, we store such a solution in the variable $BestSol$, and its preference and robustness values resp., in lb_a and lb_b. Moreover, there is a part where we test if it is possible to prune the subtree below t. If one of the pruning conditions described before is satisfied, then the Boolean variable $pruning$ is set to $true$ and a recursive call of the algorithm is performed. In this part, in the case of *PastUpperBound*, only the values $a_i(t)$ and $b_i(t)$ are used, while in the case of *PastFutureUpperBound* also $a_{ni}(t)$ e $b_{ni}(t)$ are considered. **** Algorithm 4 shows a version of our *BB* where the pruning conditions are those defined for Risky, Safe, Risky1 and Safe1, and where the upper bound is computed using *PastFutureUpperBound*.

```
Algorithm 4: BB
  input: $indexVar$: index of the current variable; $t$: partial assignment; $bestSol$: best current
         solution; $lb_a$: value of the first component of the pair associated to the best current
         solution; $lb_b$ value of the second component of the pair associated to the best current
         solution.
  if $|t| = n + 1$ then
      return;
  $var \leftarrow ObtainVariabe(indexVar)$;
  $D \leftarrow ObtainDomain(var)$;
  foreach $d \in D$ do
      $t \leftarrow t \cup (var, d)$ ;
      if $|t| = n$ then
          if $a_i(t) > lb_a$ then
              $lb_a \leftarrow a_i(t)$;   $lb_b \leftarrow b_i(t)$;   $BestSol \leftarrow (t, \langle lb_a, lb_b \rangle)$;
          else
              if $a_i(t) = lb_a$ then
                  if $b_i(t) > lb_b$ then
                      $lb_a \leftarrow a_i(t)$;   $lb_b \leftarrow b_i(t)$;   $BestSol \leftarrow (t, \langle lb_a, lb_b \rangle)$;
      else
          if $\alpha$ conditions not satisfied then
              $BB(indiceVar + 1, t, BestSol, lb_a, lb_b)$;
      $t \leftarrow t \setminus (var, d)$;
```

$*****$

9. An example

In this section we will present an example, which is similar to the one shown in (Benferhat, Dubois, Kaci, & Prade, 2006), that can be modelled by an UFCSP. We will show how to apply our solver on such a problem and how to order its solutions according to the semantics presented in Section 7.

Let us consider the work of a chair that must organize a summer school. The summer school should involve some hours of lectures, some of exercises and some of training. The chair must find an optimal partition of the hours for the various parts, which minimizes costs and takes into account the requirements of all the involved teachers and students, and several other constraints. For example, he must establish the number of training hours, knowing that they require very expensive rooms to book, and that they can contain only a certain number of students for hour, but without knowing the definite number of students that will attend them.

In the following we will present in detail the requirements that the chair has to consider. We will show that the whole problem can be modelled by an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, C_u \rangle$. Notice that we model the constraints as fuzzy constraints, since the chair wants to find the solution that maximizes the minimum preference of all the people involved in the summer school.

- The summer school must involve some hours of lectures, some hours of exercises and some hours of training. Then V_c, the set of the controllable variables that the conference chair can decide, contains variable x, which represents the number of lectures hours, variable y which stands for the exercises hours, and variable w, which represents the number of training hours.

- A requirement of the summer school is that the various part (lectures, exercises, training) must last 10, 20, or 30 hours. Hence the domains of the variables x, y and w are $D_x = D_y = D_w = \{10, 20, 30\}$.

- Training hours must be performed in rooms that can contain comfortably the students. However, the conference chair doesn't know the definite number of students that will attend them. Therefore V_u, the set of uncontrollable variables, contains variable z, that represents the number of students that will attend training hours. Moreover, the chair has invited 6 professors to teach at the summer school, but only 3 have already confirmed their participation. Thus, V_u contains also variable t representing the number of teachers with domain $D_t = \{3, 4, 5, 6\}$.

- The conference chair has received 90 student registrations. Hence, he knows that at most there will be 90 students. He believes that it's more possible that students will be between 30 and 60. In fact, he thinks that not all of them will refuse to come, since they have performed registration, but maybe not all of them will come, since the registration is free and since there are other interesting conferences in that period. This fact can be modelled giving to the uncontrollable variable z three values in its domain that are few (between 0 and 30), $average$ (between 30 and 60) and $many$ (between 60 and 90), and associating them with the following possibility distribution: $p_1(few) = 0.4$, $p_1(average) = 1$, $p_1(many) = 0.3$. As far as the number of teachers, the chair is sure that at least 3 will participate. Moreover, he believes that, while it is unlikely that all 6 of the invited professors will accept, it is quite possible that 4 or 5 will. This is modelled by the following probability distribution: $p_2(3) = 1$, $p_2(4) = 0.8$, $p_2(5) = 0.8$, $p_2(6) = 0.2$

- C_c, the set of constraints defined only on controllable variables, contains the following fuzzy constraints:

  - The lecture professor prefers to teach many hours, since he wants to explain well the material of his lessons. This can be modelled by the fuzzy constraint $c_1 = \langle f_1, \{x\} \rangle$, where $f_1(10) = 0.2$, $f_1(20) = 0.9$, and $f_1(30) = 1$.
  - The exercise professor prefers to teach a small number of hours, since he is very busy in that period. This can be described by the fuzzy constraint $c_2 = \langle f_2, \{y\} \rangle$, where $f_2(10) = 0.9$, $f_2(20) = 0.4$, and $f_2(30) = 0.1$.
  - The training hours must be done in laboratory rooms, that are expensive. Hence the conference chair, who wants to reduce costs, prefers to reduce these hours: $c_3 = \langle f_3, \{w\} \rangle$, where $f_3(10) = 0.6$, $f_3(20) = 0.5$, and $f_3(30) = 0.3$.

- An additional requirement is that the summer school lasts at most 50 hours. This can be represented by the fuzzy constraint $c_4 = \langle f_4, \{x, y, w\}\rangle$, where $f_4(x, y, w) = 1$ if $x + y + w \leq 50$, and $f_4(x, y, w) = 0$ if $x + y + w > 50$.
- The students prefer to attend many hours of lessons to learn more. This can be modelled by fuzzy constraint $c_5 = \langle f_5, \{x, y, w\}\rangle$, where $f_5(x, y, w) = 0.4$ if $x + y + w = 30$, $f_5(x, y, w) = 0.5$ if $x + y + w = 40$, and $f_5(x, y, w) = 1$ if $x + y + w = 50$.
- Moreover, the students prefer to attend many hours of exercises to ease the learning. We can model this requirement with the fuzzy constraint $c_6 = \langle f_6, \{y\}\rangle$, where $f_6(10) = 0.7$, $f_6(20) = 0.8$, and $f_6(30) = 1$.

- C_{cu}, the set of constraints involving both controllable and uncontrollable variables, contains the following fuzzy constraint:

  - The laboratory rooms, where training hours will be done, can contain comfortably 30 people. The chair wants to be able to allow every student to attend at least 10 training hours, but, since every training hour has a cost and he wants to save money, he doesn't want to book a number of training hours higher than the needed ones. Thus, if the people are few (i.e., from 0 to 30 people), he prefers to book only 10 hours, if the people are an average number (i.e., from 30 to 60 people), he prefers to book 20 hours, while if there are many people (i.e, from 60 to 90 people), he prefers to book 30 training hours. Moreover, since he wants to reduce costs, he prefers more to book a small number of training hours than a big number of them. We can model this requirement with the constraint $c_7 = \langle f_7, \{w, z\}\rangle$, where $f_7(10, few) = 1$, $f_7(10, average) = 0.5$, $f_7(10, many) = 0.1$, $f_7(20, few) = 0.1$, $f_7(20, average) = 0.9$, $f_7(20, many) = 0$, $f_7(30, few) = 0.1$, $f_7(30, average) = 0.2$, and $f_7(30, many) = 0.8$.

- C_u, the set of constraints involving only uncontrollable variables, contains constraint c_8 that models the fact that the more students will attend the school the more professors the chair would prefer to have. Thus, constraint c_8 is defined as follows: $c_8 = \langle f_8, (z, t)\rangle$,where $f_8 = (few, 3) = 1$, $f_8 = (few, 4) = 0.8$, $f_8 = (few, 5) = 0.7$,$f_8 = (few, 6) = 0.3$ $f_8 = (average, 3) = 0.6$, $f_8 = (average, 4) = 1$, $f_8 = (average, 5) = 0.9$,$f_8 = (average, 6) = 0.7$ $f_8 = (many, 3) = 0.3$, $f_8 = (many, 4) = 0.8$, $f_8 = (many, 5) = 0.9$, and $f_8 = (many, 6) = 1$.

The structure of the UFCSP which models such a problem is presented in Figure 5 (a).
We will now describe the different steps of our procedure on this example.

- *Preprocessing:* since uncontrollable variable t is not connected to any controllable variable, in this phase we connect it to w by composing $c_7 \in C_{cu}$ and $c_8 \in C_u$ and by projecting the new constraint on w and t. We call this new constraint c_9 and we note that it is defined as follows: $c_9 = \langle f_9, (w, t)\rangle$, such that $f_9(10, 3) = 1$, $f_9(10, 4) = 0.8$, $f_9(10, 5) = 0.7$, $f_9(10, 6) = 0.5$, $f_9(20, 3) = 0.6$, $f_9(20, 4) = 0.9$, $f_9(20, 5) = 0.9$, $f_9(20, 6) = 0.7$, $f_9(30, 3) = 0.3$, $f_9(30, 4) = 0.8$, $f_9(30, 5) = 0.8$, $f_9(30, 6) = 0.8$. After this step C'_{cu} will contain c_7 and c_9.

- *Enforcing generalized soft arc consistency (GSAC) on $C'_{cu} \cup C_u$:* in our example $C'_{cu} \cup C_u = \{c_7, c_8, c_9\}$. After arc consistency, the preferences induced on the values of w, namely 10,

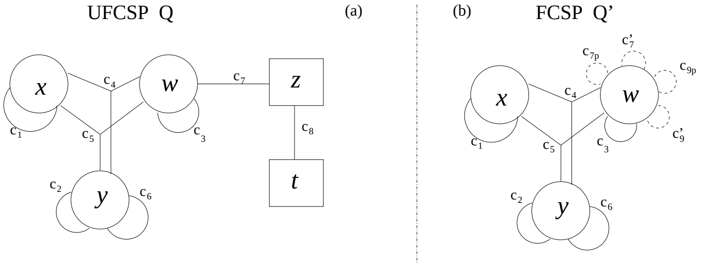


Figure 5: An UFCSP and the corresponding RFCSP obtained by applying *SP*.

20, and 30, are respectively 1, 0.9 and 0.8; those induced on the values of z, namely few, $average$ and $many$ are respectively 1, 0.9 and 0.8; while those induced on the values of t, namely 3, 4, 5, and 6 are respectively 1, 0.9, 0.9 and 0.8.

- *SP algorithm* (presented in Section 5): at this point our procedure removes uncontrollable variables z and t and all the constraints involving them, namely c_7, c_8 and c_9, by adding the robustness and projection constraints, obtained from c_7 and c_8 on w, which we will denote respectively as c_{7_p}, c_7', c_{9_p}, c_9'. The preference functions of such constraints are defined as follows: $f_{7_p}(10) = 1$, $f_{7_p}(20) = 0.9$, $f_{7_p}(30) = 0.8$, and $f_{7_p}= f_{9_p}$ while $f_7'(10) = 0.5$, $f_7'(20) = 0.6$, $f_7'(30) = 0.2$, and $f_9'(10) = 0.7$, $f_9'(20) = 0.6$, $f_9'(30) = 0.3$.

The RFCSP $Q' = \langle S_{FCSP}, V_c, C^*, C_{rob}\rangle$ obtained after SP is shown in Figure 5 (b) where:

- $C^* = C_c \cup C_{proj}$;
- $C_c = \{c_1, c_2, c_3, c_4, c_5, c_6\}$;
- $C_{proj} = \{c_{7p}, c_{9p}\}$,
- $C_{rob} = \{c_{7'}, c_{9'}\}$.

The optimal solutions for the semantics presented in Section 7 are: $s_1 = (30, 10, 10)$ for Risky and Risky-Mix semantics, $s_2 = (20, 10, 20)$ for Safe and Safe-Mix semantics, and s_1 and s_2 for Diplomatic semantics. Hence, if the conference chair uses Risky or Risky-Mix, he proposes a partition of the hours that doesn't cost very much, but is risky. In fact, if there will be large number of students, he will not be able to accommodate every student in the training hours. Whereas, if he uses Safe or Safe-Mix, he will guarantee the training hours to an average number of students. Thus he will pay laboratory rooms for more hours, even if this increases the cost. If he uses Diplomatic, he will adopt a solution that either is not expensive, or that guarantees training hours to almost all students that will come.

Summarizing, the various semantics propose different optimal solutions depending on different attitudes to risk with respect to uncertainty. This attitude is very different from the one of the Mixed semantics, that considers equally optimal solutions s_1, s_2, and $s_5 = (20, 10, 10)$, even if, as we have just shown, they have very different implications.

10. Experimental setting and results

We have implemented the solver in Java, and we have tested it on randomly generated UFCSPs, assuming that constraints in C_u are only binary, while the constraints in $C_c \cup C_{cu}$ can be also non-binary. Preferences different from zero in such constraints have been randomly generated considering a uniform distribution over the interval $(0, 1]$.

To generate such problems, we have considered the following parameters:

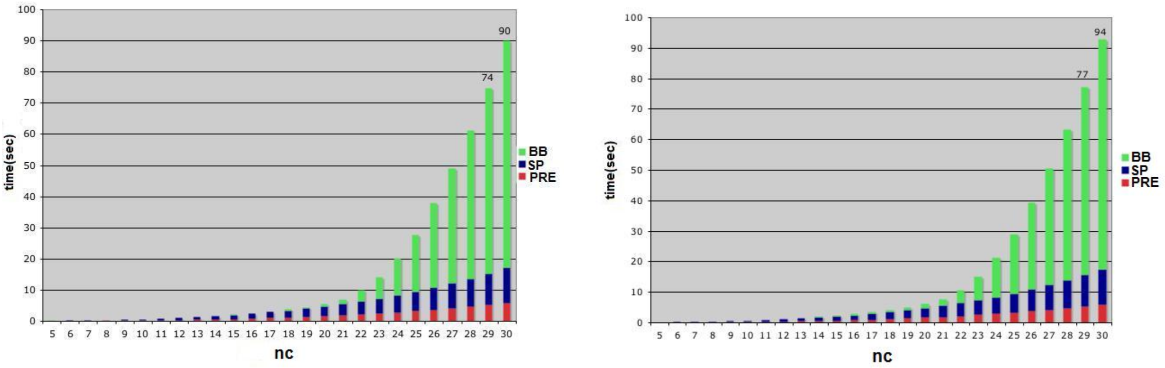
- nc: number of controllable variables;
- nu_1: number of uncontrollable variables involved in constraints of C_{cu};
- nu_2: number of further uncontrollable variables involved in constraints of $C_u \setminus C_{cu}$;
- m: cardinality of the domain of each variable;
- $d = (d_1, \ldots d_k)$ and $a = (a_1, \ldots a_k)$: d_i is the density of the constraints of arity a_i, that is, the percentage of constraints of arity a_i present in the problem w.r.t. the total number of possible constraints of that arity that can be defined on $nc + nu_1 + nu_2$ variables;
- t: tightness, that is, the percentage of tuples with preference 0 in each domain and in each constraint w.r.t. the total number of tuples.

For example, if the parameters of the generator are $nc = 3$, $nu_1 = 3$, $nu_2 = 4$, $m = 5$, $(d_1, d_2) = (50, 30)$, $(a_1, a_2) = (2, 3)$, $t = 10$, it will generate an UFCSP with 3 controllable variables, 3 uncontrollable variables which are connected to at least a controllable one, and 4 controllable variables which are not connected to any controllable variable. Since there are 10 variables, then the total number of binary constraints that can be defined is $10(10-1)/2 = 45$. Since the density of the binary constraints is 50% ($d_1 = 50$), only 22 binary constraints are generated. The total number of ternary constraints that can be defined with 10 variables is $10(10-1)(10-2)/3 = 240$. However, since the density of the ternary constraints is 30% ($d_2 = 30$), then only 72 ternary constraints are generated. Each variable has 5 elements in the domain. Thus, in the binary constraints, there are $5 * 5 = 25$ preference values and in the ternary constraints there are $5 * 5 * 5 = 125$ preference values. Since 10% of the preference values should be zero ($t = 10$), in the binary constraints 2 preference values are zero, and in the ternary constraints 12 preference values are zero, while the other preference values are randomly generated considering a uniform distribution over the interval $(0, 1]$.

In the experiments, that we will show in the following, we have generated classes of UFCSPs by varying one parameter at a time, and fixing the other ones. The varying parameters are the number nc of the controllable variables, and the density of constraints for the various arities. When the number nc of variables varies (from 5 to 30, with step 1), we set $nu_1 = 5$, $nu_2 = 5$, $m = 4$, $d = (40, 1, 1)$, $a = (2, 3, 4)$, and $t = 60$. When we vary the density (from $d = (40, 10, 10)$, $a = (2, 3, 4)$ to new values increased of 5% at every step), we set $nc = 15$, $nu_1 = 5$, $nu_2 = 5$, $m = 4$, and $t = 60$. This choice of the parameters allows us to generate problems with size 15 to 40, which are reasonably constrained.

In the experiments we have measured the time employed by the four versions of *BB*. For each fixed value of all the parameters, we show the average time over 10 different randomly generated instances. All our experiments have been performed on a computer with AMD64 processor at 2 GHz.

We started comparing the performance of *SAC* and *GSAC* when varying the number of controllable variables. Experimental results are shown in Figure 6 (a) and in Figure 6 (b). For every value of the controllable variables on the x axis, there is a bar indicating the time employed by the solver. In particular, the low part of the bar, called PRE, shows the amount of time employed by the *Preprocessing* phase and *GSAC* algorithm, the medium part of the bar, called SP, shows the amount of time employed by the *SP* algorithm, while the high part of the bar, called BB, shows the time needed by the branch and bound procedure. It is possible to see that the time needed to find an optimal solution applying *SAC* (Figure 6 (a)) and applying *GSAC* (Figure 6 (b)) are very similar.



(a)

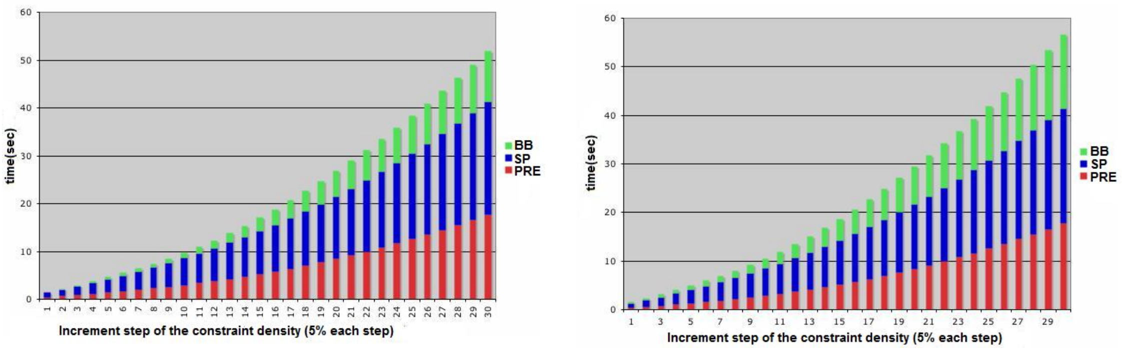
(b)

Figure 6: Comparison between *SAC* (a) and *GSAC* (b) when varying nc.

If instead we vary the constraint density, then *SAC*, shown in Figure 7 (a), is slightly better than *GSAC*, shown in Figure 7 (b), especially when the constraint density increases, and mainly in the high part of the bar, that considers the time employed by the branch and bound procedure. Note that BB is much faster than SP in Figure 7. This depends on having used a small value of n_c ($n_c = 15$). If we had done Figure 7 with a larger nc, BB would have taken more time w.r.t. SP and PRE. In fact, Figure 6 shows that BB takes increasingly more time than SAC an GSAC as the size of the problem grows. Moreover, Figure 7 shows that an increase of the density does not change the time distribution among PRE, SP, and BB.

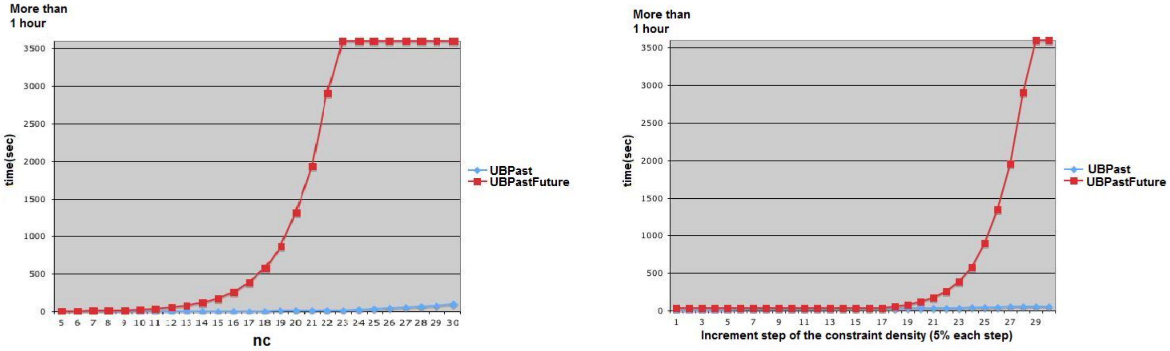
We have then considered what type of upper bound performs better, when varying the number of controllable variables and the constraint density. In both cases (see Figure 8 (a) and Figure 8 (b)) *PastUpperBound*, that does not consider future variables in the upper bound computation, performs much better than *PastFutureUpperBound*, that considers also future variables. This is probably due to the cost of constraint propagation, which must be done on non-instantiated variables for *PastFutureUpperBound*.

Finally, we have considered how the time for the overall algorithm varies in the various semantics and how it depends on the number of controllable variables (Figure 9 (a)) and on the density of the constraints (Figure 9 (b)). We have seen that Safe-Mix performs better than Risky-Mix, Risky, Safe and Diplomatic. This is probably due to the fact that Safe-Mix performs more pruning, since it considers as a first component the minimum between the preference and the robustness values, and not only one of these two values. More precisely, Safe-Mix is the best semantics, followed



(a) (b)

Figure 7: Comparison between *SAC* (a) and *GSAC* (b) when varying the constraint density.



(a) (b)

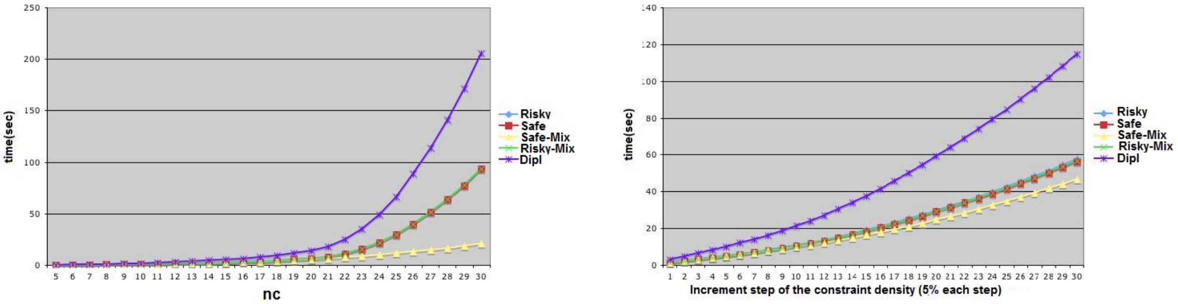
Figure 8: Comparison between *PastUpperBound* and *PastFutureUpperBound* when varying nc (a) and when varying the constraint density (b).

by Risky-Mix, Risky and Safe, both w.r.t. performance and effectiveness of pruning. We however recall that a crucial aspect for choosing the semantics in practice is the attitude towards risk it represents. The Diplomatic semantics is the worse one in both the considered cases. This could be due to its very restrictive pruning conditions, that allow only for little pruning.

Summarizing, in these experiments, the best performance of the solver can be obtained by using *SAC* as propagation algorithm, *PastUpperBound* as method for computing the upper bound, and Safe-Mix as semantics.

11. Related work

We have defined a new way for integrating fuzzy preferences and possibilistic uncertainty that assumes commensurability between preferences and possibilities scales and that satisfies some desirable properties. This new method allows us to discriminate the preference level and the robustness value of a solution, and so to obtain a solution ordering which better reflects the desirability and the robustness of a solution.



(a)

(b)

Figure 9: Comparison among Risky, Safe, Risky-Mix, Safe-Mix and Diplomatic when varying nc (a) and when varying the constraint density (b).

Another approach for handling problems with fuzzy preferences and possibilistic uncertainty, that assumes commensurability between preferences and possibilities, has been considered in (Dubois et al., 1996). In that work the notion of robustness is identical to our notion. However, the semantics they use to order the solutions does not satisfy the desired properties.

Some other effort has been done to unify preferences and possibilistic uncertainty (Amgoud & Prade, 2004) assuming commensurability. In (Amgoud & Prade, 2004) two different approaches for integrating fuzzy preferences and possibilistic uncertainty are proposed: a pessimistic and an optimistic one. However, both methods propose to mix robustness and preference level. In our framework we calculate the degree of preference of a solution combining some of the preferences of its subtuples in a pessimistic way, and the other ones in an optimistic way. In fact, given a solution s, its preference degree $pref(s)$ is obtained by combining via the minimum operator (and thus by using a pessimistic approach) $proj(s)$, that is computed in optimistic way, and the preference over all the constraints defined only on controllable variables. Hence, in computing the preference degree of a solution, we are *less pessimistic* than the pessimistic approach in (Amgoud & Prade, 2004), where all preferences of the subtuples of the solutions are calculated and combined in pessimistic way; and *less optimistic* than the optimistic approach in (Amgoud & Prade, 2004), where all the preferences of the subtuples of the solutions are calculated and combined in optimistic way.

Another approach that assumes commensurability between preferences and possibilities have been described in (Garcia & Sabbadin, 2006) in the context of the influence diagrams. Influence diagrams (Howard & Matheson, 1984) are graphical models which allow to model in a compact form problems of sequential decision making under uncertainty, where the uncertainty is specified by probability distributions over uncontrollable variables. In (Garcia & Sabbadin, 2006) the structure of influence diagrams have been extended in order to take into account possibility distributions, instead of the classical probabilities, over the uncontrollable variables. Such an extension has been performed by replacing expected utility, usually used when we have probabilities, by the pessimistic and optimistic possibilistic expected utility (Dubois & Prade, 1995). In our paper we haven't used influence diagrams to model and solve USCSPs, but we have adapted and exploited the soft constraint machinery for modelling and solving fuzzy CSPs. Moreover, we have considered possibilities only to compute the robustness degree of a solution and not also its preference degree as instead required by the possibilistic pessimistic and optimistic expected utility to be able to handle separately the certain part of the problem and the uncertain one.

Other approaches for reasoning with preferences and uncertainty are the ones in (Dubois, Fargier, & Perny, 2002; Fargier & Sabbadin, 2003; Brafman & Tennenholtz, 1996, 1997). However, they don't mix preferences and uncertainty, since they do not assume commensurability. Moreover, their approaches are based on qualitative decision theory, while our procedure is based on quantitative decision theory.

Decision problems with uncertainty have been considered, for example, in Stochastic CSPs (Walsh, 2002) and (Fargier, Lang, Martin-Clouaire, & Schiex, 1995b). However, in these approaches they assume that uncertain events are characterized by probability distributions, and not by possibility ones. In such approaches the order in which variables can be considered is not free. However, in (Pralet, Schiex, & Verfaillie, 2006b) it has been defined a formalism that allows more freedom in the elimination order, but such a formalism does not take into account possibility information. Possibility distributions and other forms of knowledge about uncertainty have been instead considered in (Pralet, Schiex, & Verfaillie, 2006a) that provides a generic algebraic framework for modelling sequential decision making with uncertainties, feasibilities and utilities, where partial observability is assumed. In (Pralet et al., 2006a) the main focus is on the modelling phase than in determining various notions of optimality of the solutions like in our paper. In (Pralet et al., 2006a) an approach to solve problems with uncertainty under the form of possibilistic information uses decision trees and it exploits possibilistic pessimistic and optimistic expected utility, therefore it is different from our proposal, since we don't use decision trees and since, as explained before, we don't consider classical possibilistic pessimistic and optimistic expected utility to handle possibilities.

Uncertainty and constraints have also been considered in (Freuder & Wallace, 1998), where dynamic constraint satisfaction problems have been analyzed. Such problems are CSPs that change intermittently over time, by loss or gain of values, variables or constraints. In such a context the goal is to find methods for finding solutions that are stable, i.e., solutions that are more likely to remain valid after changes that temporarily alter the set of valid assignments. The setting considered in (Freuder & Wallace, 1998) is thus different from the one that we considered in this paper, since they don't distinguish between controllable and uncontrollable variables and since they don't have a fixed set of constraints, variables and domain, but only a set of variables, constraints and domains that may change over time. The concept of stability is different from our concept of robustness. In fact, a solution in (Freuder & Wallace, 1998) is stable if it remains valid for the altered problem, while in our paper a solution is robust if it guarantees an high level of preferences in the uncertain part of the problem for a big number of scenarios.

12. Summary and future work

In this paper we have defined UFCSPs, i.e., problems with fuzzy preferences and uncontrollable variables with a possibility distribution over their domains. We have defined the notion of preference and robustness for the solutions of such problems, as well as properties that such notions should respect, also in relation to the solution ordering.

We have then provided an algorithm to remove the uncontrollable part of the problem, while altering the controllable part in order to loose little information. On the resulting problem, we have then defined the preference and robustness of a solution of the initial UFCSP. Different semantics use such two notions to order the solutions according to different attitudes to risk. We have then discussed how our proposed notions of preference and robustness, as well as our semantics, satisfy the desired properties.

Following this approach, we have defined a solver for UFCSPs, that first removes the uncontrollable part of the problem, and then finds an optimal solution of the controllable part according to a chosen semantics. To develop such a solver, we have adapted constraint propagation and branch and bound techniques to our framework. Finally, we have tested the solver on randomly generated UFSCPs.

In this paper we have focused on problems with fuzzy preferences and uncertainty. We plan to extend and formalize the research to other classes of soft constraints, not necessarily fuzzy. This is partially presented in {removed for anonymity}. Moreover, we plan to extend our framework to problems with bipolar preferences (Bistarelli, Pini, Rossi, & Venable, 2006, 2007a, 2008). A first step in this research has been shown in (Bistarelli, Pini, Rossi, & Venable, 2007b), and states that the properties hold also in the bipolar context, when we consider sets of preferences that are totally ordered.

In our paper we have handled uncertain problems by assuming to have no observability of uncertain events before decision. We intend to analyze also the assumption of full observability of uncertain events just before decision. This has been done for uncertain problems with uncontrollable variables specified by probability distributions in (Fargier et al., 1995b). We intend to reason on it in our framework where the uncontrollable variables are characterized by possibility distributions.

Acknowledgment

We would like to thank S.A. and M. R. (abbreviated for anonymity) for their relevant contribution to the experimental part of this paper.

Appendix

In the following we will sometimes need to use a preference value that we call controllable preference. More precisely, we will denote the controllable preference of a solution s with $control(s) = min_{\{\langle def,con\rangle \in C_c\}}\ def(s\downarrow_{con})$.

Proposition 1 *The definition of robustness given in Definition 16 satisfies properties 1 and 2.*

Proof: We first show that the definition of robustness given in Definition 16 satisfies Property 1.

Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi\ C_c,\ C_{cu}, \emptyset\rangle$, where, for every constraint, $c^i = \langle f^i, con^i\rangle$, such that $con^i \cap V_c = X^i$ and $con^i \cap V_u = Z^i$, with possibility distribution π_{Z_i}, let, $f^i(d,a) \leq f^i(d',a)$, for all a assignments to Z_i and for d and d' assignments to X^i. To prove Property 1 we will show that, given two solutions s and s' of Q, such that $s \downarrow_{X^i} = d$ and $s' \downarrow_{X^i} = d'$, then $rob(s) \leq rob(s')$. We recall that, for every constraint $c^i = \langle f^i, con^i\rangle$ in the statement of Proposition 1, $f'^i(d) = min_{a\in A_{z^i}}\ max(f^i(d,a), r(\pi_{Z^i}(a)))$ and $f'^i(d') = min_{a \in A_{z^i}}$ $max(f^i(d',a), r(\pi_{Z^i}(a)))$, where A_{z^i} is the Cartesian product of the domains of the variables in Z^i. Since $f^i(d,a) \leq f^i(d',\ a)$, $\forall a$, then $max(f^i(d,a),\ r(\pi_{Z^i}(a))) \leq max(f^i(d',\ a),\ r(\pi_{Z^i}(a)))$, $\forall a$. Therefore, $min_{a\in A_{z^i}}\ max(f^i_1(d,a),\ r(\pi_{Z^i}(a))) \leq max(f^i(d,a),\ r(\pi_{Z^i}(a))) \leq max\ (f^i(d',$ $a),\ r(\pi_{Z^i}(a)))$, $\forall a$. This allows to conclude that, since $s \downarrow_{X^i} = d$ and $s' \downarrow_{X^i} = d'$, $f'^i(s \downarrow_{X^i})$ $= min_{a\in A_{z^i}}\ max\ (f^i(s \downarrow_{X^i}, a),\ r(\pi_{Z^i}(a))) \leq min_{a\in A_{z^i}}\ max(f^i(s' \downarrow_{X^i}, a),\ r(\pi_{Z^i}(a))) =$

$f'^i(s' \downarrow_{X^i})$. The fact that $rob(s) = min_i f'^i(s \downarrow_{X_i})$ and $rob(s') = min_i f'^i(s' \downarrow_{X_i})$ allows us to conclude.

We now show that the definition of robustness given in Definition 16 satisfies Property 2. Consider two UFCSPs: $Q_1 = \langle S_{FCSP}, V_c, V_u, \pi^1, C_c, C_{cu}, \emptyset\rangle$ and $Q_2 = \langle S_{FCSP}, V_c, V'_u, \pi^2, C_c, C_{cu}, \emptyset\rangle$, where V_u and V'_u are the same set of uncontrollable variables described, however, by different possibility distributions, that we denote respectively with π^1 and π^2. In particular, for every constraint, $c^i = \langle f^i, con^i\rangle$, such that $con^i \cap V_c = X^i$ and $con^i \cap V_u = Z^i$, let $\pi^1_{Z^i}(a) \geq \pi^2_{Z^i}(a)$, for all a assignments to Z^i. To prove Property 2 we will show that, given solution s of Q_1 and Q_2, such that $s \downarrow_{X^i} = d$, we have $rob_1(s) \leq rob_2(s)$, where rob_i is the robustness computed in the problem Q_i. As above, we have that, for every constraint, $f'^i_1(d) = min_{a \in A_{z^i}}$ $max(f^i(d,a), r(\pi^1_{Z^i}(a)))$ and $f'^i_2(d) = min_{a \in A_{z^i}}$ $max(f^i(d,a), r(\pi^2_{Z^i}(a)))$. Moreover, since r is an order-reversing map, if $\pi^1_{Z^i}(a) \geq \pi^2_{Z^i}(a)$ then $r(\pi^1_{Z^i}(a)) \leq r(\pi^2_{Z^i}(a))$, $\forall a$. Thus, $max(f^i(d,a),$ $r(\pi^1_{Z^i}(a))) \leq max(f^i(d,a), r(\pi^2_{Z^i}(a)))$, $\forall a$. From here we can conclude as above. □

Proposition 2 *Consider two solutions of an UFCSP, say s_1 and s_2. If $s_1 >_{Mixed} s_2$, then either $s_1 >_D s_2$ or $s_1 \bowtie_D s_2$. Also, $s_1 >_{R1,S1} s_2$.*

Proof: Let us show the first item. Assume, for the sake of contradiction, that $s_1 <_D s_2$. Thus it must be that either $pref(s_1) \leq pref(s_2)$ and $rob_{(}s_1) < rob(s_2)$, or $pref(s_1) < pref(s_2)$ and $rob_{(}s_1) \leq rob(s_2)$. In both cases, $min(pref(s_1), rob(s_1)) \leq min(pref(s_2), rob(s_2))$, which is in contradiction with $s_1 >_{Mixed} s_2$.

Let us show the second item. Since $s_1 >_{Mixed} s_2$, $min(pref(s_1), rob(s_1)) > min(pref(s_2),$ $rob(s_2))$ and thus, $s_1 >_{R1,S1} s_2$. □

Proposition 3 *The preferential orderings over the solutions of an UFCSP induced by Risky, Risky-Mix, Safe, Safe-Mix and Diplomatic do not satisfy Property 3.*

Proof: Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset\rangle$. Given two solutions s and s' of Q, i.e., assignments to V_c, we now show that, if $\forall a_i$ assignments to V_u in Q, $ovpref(s, a_i) > ovpref(s', a_i)$, it can happen that $s \not>_J s'$, where $J = R, R1, S, S1, D$.

- *Risky and Risky-Mix.* For these semantics it can happen that $s \not> s'$.

  Let us consider an UFCSP $Q = \langle S_{FCSP}, V_c, \pi, V_u, C_c, C_{cu}, \emptyset\rangle$ where $V_c = \{x\}$, $V_u = \{z, y, t\}$, the set C_c is empty and the set C_{cu} is composed by two constraints: $c_1 =< f_1, \{x, y, t\} >$ and $c_2 =< f_2, \{x, y, z\} >$. Let us assume that the domain of x is $\{s, s'\}$, the domain of y, t and z is $\{0, 1\}$, and that the preference functions of c_1 and c_2 are defined as follows: $f_1(x = s', y = b, t = c) = 1$ if $b = c = 0$, $f_1(x = s', y = b, t = c) = 0$ otherwise, $f_1(x = s, y = b, t = c) = 0.5$ for all b and c in $\{0, 1\}$, $f_2(x = s', y = b, z = c) = 1$ if $b = c = 1$, $f_2(x = s', y = b, z = c) = 0$ otherwise, $f_2(x = s, y = b, z = c) = 0.5$ for all b and c in $\{0, 1\}$. Moreover, let us assume that the joint possibility on y and t is such that $\pi_{yt}(y = b, t = c) = 1$ if $b = c = 0$, $\pi_{yt}(y = b, t = c) = 0.5$ otherwise, and that the joint possibility on y and z is such that $\pi_{yz}(y = b, z = c) = 1$ if $b = c = 1$, $\pi_{yz}(y = b, z = c) = 0.5$ otherwise. Then the overall preferences are: $ovpref(x = s', y = b, z = c, t = d) = 0$ for all b,c, and d in $\{0, 1\}$, and $ovpref(x = s, y = b, z = c, t = d) = 0.5$ for all b,c, and d in $\{0, 1\}$, i.e., $ovpref(x = s, y = b, z = c, t = d) > ovpref(x = s', y = b, z = c, t = d)$ for all b,c, and d in $\{0, 1\}$, hence the solutions s and s' satisfy the hypothesis. The robustness values for the solutions s and s' are $rob(s) = rob(s') = 0.5$ and the preference degrees are $pref(s') = 1$ and $pref(s) = 0.5$. Therefore, $s' >_R s$ for Risky, since $pref(s') > pref(s)$, and also $s' >_{R1} s$ for Risky-Mix since $min(pref(s), rob(s)) = min(pref(s'), rob(s'))$ and $pref(s') > pref(s)$.

- *Safe and Diplomatic.* For these semantics it can happen that $s \not> s'$.

  In fact, let us consider an UFCSP $Q = \langle S_{FCSP}, V_c, \pi, V_u, C_c, C_{cu}, \emptyset\rangle$ where $V_c = x$, $V_u = z$, C_c is composed by $c_1 =< f_1, \{x\} >$, C_{cu} by $c_2 =< f_2, \{x, z\} >$, and where $D_z = \{a_1, a_2\}$ and $D_x = \{s, s'\}$ are respectively the domain of z and x. Let us assume that the possibility distribution on z is such that $\pi(a_1) = 1$ and $\pi(a_2) = 0.7$. Let us assume moreover that $f_2(s, a_1) = 0.4$, $f_2(s, a_2) = 0.5$, $f_2(s', a_1) = 0.8$, $f_2(s', a_2) = 0.9$, $f_1(s) = 0.3$ and $f_1(s') = 0.2$. Then the overall preferences are: $ovpref(s, a_1) = 0.3$, $ovpref(s, a_2) = 0.3$, $ovpref(s', a_1) = 0.2$, $ovpref(s', a_2) = 0.2$, i.e., $ovpref(s, a_i) > ovpref(s', a_i), \forall a_i$, $i = 1, 2$, hence s and s' satisfy the hypothesis. The robustness values for s' and s are $rob(s) = 0.4$, $rob(s') = 0.8$ and the preference degrees are $pref(s) = 0.3$ and $pref(s') = 0.2$. Therefore, $s <_S s'$ for Safe semantics, and $s \bowtie_D s'$ for Diplomatic semantics.

- *Safe-Mix.* For this semantics it can happen that $s \not> s'$.

  In fact, let us consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset\rangle$ where $V_c = x$, $V_u = z$, C_c is composed by $c_1 =< f_1, \{x\} >$, C_{cu} is composed by $c_2 =< f_2, \{x, z\} >$, and where $D_z = \{a_1, a_2\}$ and $D_x = \{s, s'\}$ are respectively the domain of z and x. Let us assume that the possibility distribution on z is such that $\pi(a_1) = 1$ and $\pi(a_2) = 0.7$. Let us assume moreover that $f_2(s, a_1) = 0.5$, $f_2(s, a_2) = 0.2$, $f_2(s', a_1) = 0.4$, $f_2(s', a_2) = 0.1$, $f_1(s) = 0.9$ and $f_1(s') = 0.9$. Then the overall preferences are: $ovpref(s, a_1) = 0.5$, $ovpref(s, a_2) = 0.2$, $ovpref(s', a_1) = 0.4$, $ovpref(s', a_2) = 0.1$, i.e., $ovpref(s, a_i) > ovpref(s', a_i), \forall a_i$, $i = 1, 2$, hence s and s' satisfy the hypothesis. The robustness values for s' and s are $rob(s) = 0.3$, $rob(s') = 0.3$ and the preference degrees are $pref(s) = 0.5$ and $pref(s') = 0.4$. Since $min(pref(s), rob(s)) = 0.3 = min(pref(s'), rob(s'))$ and $rob(s) = rob(s')$, then $s =_{S1} s'$ for Safe-Mix semantics.

Proposition 4 *Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset\rangle$. If we compute the constraint obtained by combining all the constraints in C_{cu} and, starting directly from such a constraint, we compute the projection constraint over the controllable variables of C_{cu} and the robustness constraint, then the preferential orderings over the solutions of Q induced by Risky and Risky-Mix satisfy Property 3.*

Proof: To prove the proposition we will show that, given two solutions s and s' of Q, i.e., assignments to V_c, if $\forall a_i$ assignments to V_u in Q, $ovpref(s, a_i) > ovpref(s', a_i)$, then $s >_J s'$, where $J = R, R1$.

- *Risky-Mix.* Assume to combine all the constraints of the set C_{cu}. We thus obtain a single constraint over the variables in C_{cu}. We denote with C_2 such a constraint and with f_2 its preference function. Assume also to project the constraint C_2 over the controllable variables of C_{cu} and denote with C_1 the obtained constraint, and with f_1 its preference function. Assume also to combine all the constraints in C_c and denote such a constraint with C_3 and with f_3 its preference function. Assume to associate to the set of uncontrollable variables of C_2, i.e., to the variables in V_u, a possibility distribution, that we call π, obtained by performing the joint possibility (see Section 5) of all the possibility distributions of the variables in V_u. Given an assignment s to all the variables of V_c, its preference on constraint C_1 is $f_1(s) = control(s)$, on C_3 is $f_3(s) = proj(s)$ and on $C_1 \bigotimes C_3$ is $min(f_1(s), f_3(s)) = min(control(s), proj(s)) = pref(s)$. Given an assignment (s, a_i), where a_i is a complete assignment to variables of V_u, its preference $f_2(s, a_i)$, is obtained performing the minimum of the preferences associated to all the subtuples of (s, a_i) by the constraints of Q involving at least one variable in V_u and one in V_c. Using this new notation we have that $\forall s, a_i$ assignments to the variables in V_c and V_u, $ovpref(s, a_i) = min(f_1(s), f_2(s, a_i)) = min(control(s), f_2(s, a_i))$.

We want to show that if $ovpref(s, a_i) > ovpref(s', a_i), \forall a_i$, then $s{>_{R1}}\ s'$, i.e., $min(pref(s), rob(s)) > min(pref(s'), rob(s'))$ or $(min(pref(s), rob(s)) = min(pref(s'), rob(s'))$ and $pref(s) > pref(s'))$. First, we show that if $ovpref(s, a_i) > ovpref(s', a_i)$, $\forall a_i$, then $pref(s) > pref(s')$. Then, for proving $s >_{R1} s'$, it is sufficient to prove that $min(pref(s), rob(s)) \geq min(pref(s'), rob(s'))$.

First part. If $ovpref(s, a_i) > ovpref(s', a_i)$, $\forall a_i$ assignment to the variables in V_u, then $pref(s) > pref(s')$.

In fact, if $ovpref(s, a_i) > ovpref(s', a_i), \forall a_i$, then this holds also for a_{i*} such that $proj(s') = f_2(s, a_{i*})$. Then we have $min(control(s),\ f_2(s, a_{i*})) > min(control(s'),\ f_2(s', a_{i*})) = min(control(s'),\ proj(s')) = pref(s')$. Since $proj(s) \geq f_2(s, a_{i*}))$, then $pref(s) = min(control(s), proj(s)) \geq min(control(s),\ f_2(s, a_{i*})) > pref(s')$, and so $pref(s) > pref(s')$.

Notice that from the result above follows that: if $pref(s) \leq pref(s')$, then $ovpref(s, a_i) \leq ovpref(s', a_i)$, $\exists a_i$.

Second part. If $ovpref(s, a_i) > ovpref(s', a_i)$, $\forall a_i$ assignment to the variables in V_u, then we have that $min(pref(s), rob(s)) \geq min(pref(s'), rob(s'))$.

The proof is given by contradiction. That is, we will show that if $min(pref(s), rob(s)) < min(pref(s'), rob(s'))$, then there is an assignment $a_{\bar{1}}$ such that $ovpref(s, a_{\bar{1}}) \leq ovpref(s', a_{\bar{1}})$.

1. Assume $min(pref(s), rob(s)) = pref(s)$ and $min(pref(s'), rob(s')) = pref(s')$. Since we are assuming $min(pref(s), rob(s)) < min(pref(s'), rob(s'))$, then it must be $pref(s) < pref(s')$. Then we can conclude by the first part of the proof.
2. Assume $min(pref(s), rob(s)) = pref(s)$ and $min(pref(s'), rob(s')) = rob(s')$. Then, $rob(s') \leq pref(s')$. Since we are assuming that $pref(s) < rob(s')$, then, this implies that $pref(s) < pref(s')$. Hence we can conclude as in the previous step.
3. Assume $min(pref(s), rob(s)) = rob(s)$ and $min(pref(s'), rob(s')) = rob(s')$. Moreover, and without loss of generality[3], let us consider the case in which $rob(s) < pref(s)$, $rob(s') < pref(s')$, and thus, $rob(s) < rob(s') < pref(s')$.

[3] We consider only the case with strict inequalities since if $rob(s) = pref(s)$, $rob(s') = pref(s')$ we are in Case 1, if $rob(s) = pref(s), rob(s') < pref(s')$ we are in Case 2, and if $rob(s) < pref(s), rob(s') = pref(s')$ we are in Case 4.

If we consider the UFSCP Q, then $rob(s) = min_{a_i}\ (max(f_2(s,\ a_i),\ r(\pi(a_i))))$ and $rob(s') = min_{a_i}\ (max(f_2(s', a_i), r(\pi(a_i))))$. For the sake of notation we will indicate $max(f_2(s, a_i), r(\pi(a_i)))$ (respectively $max(f_2(s', a_i), r(\pi(a_i)))$ with m_i (respectively m'_i). Let $a_{\bar{\imath}}$ and $a_{\bar{\jmath}}$ be the values for the variables in V_u such that $rob(s) = m_{\bar{\imath}}$ and $rob(s') = m'_{\bar{\jmath}}$ (i.e., $m'_{\bar{\jmath}} = max(f_2(s', a_{\bar{\jmath}}), r(\pi(a_{\bar{\jmath}})))$. Then $rob(s) = m_{\bar{\imath}} \geq f_2(s, a_{\bar{\imath}})$. Thus, since $pref(s) > rob(s)$, then $control(s) > pref(s) > rob(s) = max(f_2(s, a_{\bar{\imath}}), r(\pi(a_{\bar{\imath}}))) \geq f_2(s, a_{\bar{\imath}})$. This allows us to conclude that

$$ovpref(s, a_{\bar{\imath}}) = min(control(s), f_2(s, a_{\bar{\imath}})) = f_2(s, a_{\bar{\imath}}).$$

We will now show that, for the assignment $a_{\bar{\imath}}$, we have $ovpref(s, a_{\bar{\imath}}) < ovpref(s', a_{\bar{\imath}})$. In order to do that we will consider all the possible cases from which $m_{\bar{\imath}}$ and $m'_{\bar{\imath}}$ can derive, where $m'_{\bar{\imath}} = max(f_2(s', a_{\bar{\imath}}), r(\pi(a_{\bar{\imath}})))$.

First of all, since $m_{\bar{\imath}} = rob(s) < rob(s') = m'_{\bar{\jmath}}$ and since $m'_{\bar{\jmath}} \leq m'_{\bar{\imath}}$, then it must be $m_{\bar{\imath}} < m'_{\bar{\imath}}$. The cases to be considered are the following:

- $m_{\bar{\imath}} = m'_{\bar{\imath}} = r(\pi(a_{\bar{\imath}}))$. This can never occur since it contradicts $m_{\bar{\imath}} < m'_{\bar{\imath}}$.
- $m_{\bar{\imath}} = f_2(s, a_{\bar{\imath}})$ and $m'_{\bar{\imath}} = f_2(s', a_{\bar{\imath}})$. Thus, $f_2(s, a_{\bar{\imath}}) = m_{\bar{\imath}} < m'_{\bar{\imath}} = f_2(s', a_{\bar{\imath}})$.
  - If $m'_{\bar{\imath}} = f_2(s', a_{\bar{\imath}}) < control(s')$, then $ovpref(s', a_{\bar{\imath}}) = f_2(s', a_{\bar{\imath}})$. Since we know that $ovpref(s, a_{\bar{\imath}}) = f_2(s, a_{\bar{\imath}})$, then $ovpref(s, a_{\bar{\imath}}) < ovpref(s', a_{\bar{\imath}})$.
  - If $m'_{\bar{\imath}} = f_2(s', a_{\bar{\imath}}) \geq control(s')$, then $ovpref(s', a_{\bar{\imath}}) = control(s')$ and $ovpref(s, a_{\bar{\imath}}) = f_2(s, a_{\bar{\imath}}) = m_{\bar{\imath}} = rob(s) < m'_{\bar{\jmath}} = rob(s') < pref(s') \leq control(s') = ovpref(s', a_{\bar{\imath}})$. Again $ovpref(s,\ a_{\bar{\imath}}) < ovpref(s',\ a_{\bar{\imath}})$.
- $m_{\bar{\imath}} = f_2(s, a_{\bar{\imath}})$ and $m'_{\bar{\imath}} = r(\pi(a_{\bar{\imath}}))$. This case can never occur since it would give the following contradiction: $r(\pi(a_{\bar{\imath}})) \leq m_{\bar{\imath}} = f_2(s, a_{\bar{\imath}}) < m'_{\bar{\imath}} = r(\pi(a_{\bar{\imath}}))$.
- $m_{\bar{\imath}} = r(\pi(a_{\bar{\imath}}))$ and $m'_{\bar{\imath}} = f_2(s', a_{\bar{\imath}})$. Thus we have, $ovpref(s, a_{\bar{\imath}}) = min(control(s), f_2(s, a_{\bar{\imath}})) \leq min(control(s), r(\pi(a_{\bar{\imath}}))) = min(control(s), rob(s)) =$ (since $proj(s) \geq rob(s)$) $min(control(s), proj(s), rob(s)) = min(pref(s), rob(s)) < min(pref(s'),\ rob(s')) \leq min(pref(s'),\ f_2(s',\ a_{\bar{\imath}})) \leq min(control(s'),\ f_2(s',\ a_{\bar{\imath}})) = ovpref(s',\ a_{\bar{\imath}}))$.

4. Assume $min(pref(s), rob(s)) = rob(s)$ and $min(pref(s'), rob(s')) = pref(s')$.

   Again, we consider only the case with strict inequalities ($rob(s) < pref(s), pref(s') < rob(s')$) since all the others can be treated as one of the previous cases.

   Since $rob(s) < pref(s')$, then we have $rob(s) < pref(s') < rob(s')$. Let $m_{\bar{\imath}} = rob(s)$ and $m'_{\bar{\jmath}} = rob(s')$ as in Case 3. Since $rob(s) < pref(s) \leq control(s)$, then, as before, $ovpref(s, a_{\bar{\imath}}) = f_2(s, a_{\bar{\imath}})$.

   We will show that $ovpref(s, a_{\bar{\imath}}) < ovpref(s', a_{\bar{\imath}})$. As in the previous case, in order to do so, we consider all the possible cases from which $m_{\bar{\imath}}$ and $m'_{\bar{\imath}}$ can derive. First, notice that from $rob(s) < rob(s')$ we get $m_{\bar{\imath}} < m'_{\bar{\jmath}}$. The cases to be considered are the following:

   - $m_{\bar{\imath}} = m'_{\bar{\imath}} = r(\pi(a_{\bar{\imath}}))$. We conclude as in the corresponding step of Case 3.
   - $m_{\bar{\imath}} = f_2(s, a_{\bar{\imath}})$ and $m'_{\bar{\imath}} = f_2(s', a_{\bar{\imath}})$. Then we have $ovpref(s, a_{\bar{\imath}}) = f_2(s, a_{\bar{\imath}}) = m_{\bar{\imath}} = rob(s) < pref(s') < rob(s') = m'_{\bar{\jmath}} \leq m'_{\bar{\imath}} = f_2(s', a_{\bar{\imath}})$. Hence it must be that: $ovpref(s', a_{\bar{\imath}}) = min(control(s'), f_2(s', a_{\bar{\imath}})) \geq min(pref(s'),\ f_2(s',\ a_{\bar{\imath}})) = pref(s') > ovpref(s, a_{\bar{\imath}})$.

  - $m_{\bar{1}} = f_2(s, a_{\bar{1}})$ and $m'_{\bar{1}} = r(\pi(a_{\bar{1}}))$. We conclude like in the corresponding step of Case 3.
  - If $m_{\bar{1}} = r(\pi(a_{\bar{1}}))$ and $m'_{\bar{1}} = f_2(s', a_{\bar{1}})$, then $ovpref(s, a_{\bar{1}}) = f_2(s, a_{\bar{1}}) \leq m_{\bar{1}} = rob(s) < pref(s') < rob(s') \leq m'_{\bar{1}} = f_2(s', a_{\bar{1}})$. Hence $ovpref(s', a_{\bar{1}}) = min(control(s'), f_2(s', a_{\bar{1}})) \geq min(pref(s'), f_2(s', a_{\bar{1}})) = pref(s') > ovpref(s, a_{\bar{1}})$.

- *Risky.* We can conclude that $s >_R s'$ for Risky semantics, by using the first part of the proof for Risky-Mix semantics.

Proposition 5 *The preferential orderings over the solutions of an UFCSP induced by Risky, Risky-Mix, Safe and Diplomatic satisfy Property 4, while the one induced by Safe-Mix does not satisfy Property 4.*

Proof: Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset \rangle$. To prove the proposition we will show that, given two solutions s and s' of Q, if $rob(s) = rob(s')$ and $pref(s) > pref(s')$, then $s >_J s'$, where $J = R, S, D, R1$. Instead, it could happen that $s \not>_{S1} s'$.

- *Risky*, *Safe* and *Diplomatic* satisfy this property by definition.

- *Risky-Mix* satisfies this property. $s >_{R1} s'$ means that $min(pref(s), rob(s)) > min(pref(s'), rob(s'))$, or that $min(pref(s), rob(s)) = min(pref(s'), rob(s'))$ and $pref(s) > pref(s')$. Since $pref(s) > pref(s')$ and $rob(s) = rob(s')$ then $min(pref(s), rob(s)) \geq min(pref(s'), rob(s'))$. If we have $min(pref(s), rob(s)) > min(pref(s'), rob(s'))$, then we conclude immediately. If $min(pref(s), rob(s)) = min(pref(s'), rob(s'))$, we conclude by observing that $pref(s) > pref(s')$.

- *Safe-Mix.* In this case it can happen $s \not>_{S1} s'$. Let us recall that $s >_{S1} s'$ means that either $min(pref(s), rob(s)) > min(pref(s'), rob(s'))$, or that $min(pref(s), rob(s)) = min(pref(s'), rob(s'))$ and $rob(s) > rob(s')$. Assume, for example, that s and s' are such that $proj(s) = 0.9$, $control(s) = 0.8$ and $rob(s) = 0.5$ and $proj(s') = 0.9$, $control(s') = 0.7$ and $rob(s') = rob(s) = 0.5$. Then, s and s' satisfy the hypothesis since $pref(s) = min(proj(s) = 0.9, control(s) = 0.8) = 0.8 > pref(s') = min(proj(s') = 0.9, control(s') = 0.7,) = 0.7$. However, $s =_{S1} s'$, since $min(pref(s) = 0.8, rob(s) = 0.5) = 0.5$ is equal to $min(control(s') = 0.7, rob(s') = 0.5)$ and $rob(s') = rob(s) = 0.5$. □

Proposition 6 *The preferential orderings over the solutions of an UFCSP induced by Risky, Safe, Safe-Mix and Diplomatic satisfy Property 5, while the one induced by Risky-Mix does not satisfy Property 5.*

Proof: Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset \rangle$. To prove the proposition we will show that, given two solutions s and s' of Q, if $pref(s) = pref(s')$ and $rob(s) > rob(s')$, then $s >_J s'$, where $J = R, S, D, S1$. Instead, it could happen that $s \not>_{R1} s'$.

- *Risky*, *Safe* and *Diplomatic* satisfy this property by definition.

- *Safe-Mix* satisfies this property. If $pref(s) = pref(s')$ and $rob(s) > rob(s')$ then it must be that $min(pref(s), rob(s)) \geq min(pref(s'), rob(s'))$. If $min(pref(s), rob(s)) > min(pref$ (s'), $rob(s'))$, then we conclude immediately. If min $(pref(s)$, $rob(s)) = min(pref$ (s'), $rob(s'))$ we conclude by observing that $rob(s) > rob(s')$.

- In *Risky-Mix* it can happen that $s \not>_{R1} s'$. Consider for example solutions s and s' such that $proj(s) = 0.9$, $control(s) = 0.5$ and $rob(s) = 0.8$ and $proj(s') = 0.8$, $control(s') = 0.5$ and $rob(s') = 0.7$. We have that $rob(s) > rob(s')$ and $pref(s) = min(proj(s) = 0.9, control(s) = 0.5) = 0.5 = pref(s') = min(proj(s') = 0.8, control(s') = 0.5)$. However, since $min(pref(s) = 0.5, rob(s) = 0.8) = 0.5 = min(pref(s') = 0.5, rob(s') = 0.7)$ and $pref(s) = pref(s') = 0.5$, then $s =_{R1} s'$. □

Theorem 1 *For every solution s of an UFCSP,* $rob(s) \leq proj(s)$.

Proof: Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi, C_c, C_{cu}, \emptyset \rangle$ where $C_{cu} = \bigcup_i \langle f^i, con^i \rangle$, such that $con^i \cap V_c = X^i$ and $con^i \cap V_u = Z^i$, with possibility distribution π_{Z^i} and domain A_{Z_i}. To prove this proposition we will show that for every solution s of Q, i.e., for every assignment to X^i, we have $rob(s) \leq proj(s)$.

We recall that $proj(s) = min_i proj_i(s)$ and $proj_i(s) = max_{a \in A_{Z^i}} f^i(s, a)$. We recall also that $rob(s) = min_i f'^i(s)$, where for every constraint $c^i = \langle f^i, con^i \rangle$, $f'^i(s) = min_{a \in A_{z^i}}$ $max(f^i(s,$ $a)$, $r(\pi_{Z^i}(a)))$. By the definition of $f'^i(s)$, $f'^i(s) \leq max(f^i(s, a)$, $r(\pi_{Z^i}(a)))$, $\forall a$, and so this holds also for a such that $\pi_{Z^i}(a) = 1$. Let us call this a as $\bar{a}$. For such $\bar{a}$ we have $max(f^i(s, \bar{a})$, $r(\pi_{Z^i}(\bar{a}))) = max(f^i(s, \bar{a})$, $c(1)) = max(f^i(s, \bar{a})$, $0) = f^i(s, \bar{a})$. Therefore we have $f'^i(s) \leq f^i(s, \bar{a}) \leq proj_i(s)$, by the definition of $proj_i(s)$. The fact that $rob(s) = min_i f'^i(s)$ and that $proj(s) = min_i proj_i(s)$ allows us to conclude. □

Proposition 7 *The preferential ordering over the solutions of an UFCSP induced by Mixed semantics does not satisfy Properties 3, 4 and 5.*

Proof: Consider an UFCSP $Q = \langle S_{FCSP}, V_c, V_u, \pi\ C_c, C_{cu}, \emptyset \rangle$.

We first show that Property 3 is not satisfied. To prove this proposition we will show that, given two solutions s and s' of Q, i.e., assignments to V_c, if $\forall a$ assignments to V_u, $ovpref(s, a) > ovpref(s', a)$, then it could happen that $s \not>_{Mixed} s'$. For showing this we can use the same example considered in the proof of Proposition 4 for Safe-Mix semantics.

We now prove that Property 4 is not satisfied. To prove the proposition we will prove that, given two solutions s and s' of Q, if $rob(s) = rob(s')$ and $pref(s) > pref(s')$, then it could happen that $s \not>_{Mixed} s'$. Let us consider any pair of solutions s and s' such that $pref(s) > pref(s') > rob(s') = rob(s)$. Since, by Theorem 1, $\forall s$, $min(pref(s), rob(s)) = min(control(s), rob(s))$, and since $min(pref(s), rob(s)) = min(pref(s'), rob(s')$, then $s =_{Mixed} s'$.

We now show that Property 5 is not satisfied. To prove the proposition we will show that, given two solutions s and s' of Q, if $pref(s) = pref(s')$ and $rob(s) > rob(s')$, then it could happen that $s \not>_{Mixed} s'$. Let us consider for example the solutions s and s' satisfying the hypothesis, such that $proj(s) = 0.9$, $control(s) = 0.3$ and $rob(s) = 0.4$ and $proj(s') = 0.3$, $control(s') = 0.3$ and $rob(s') = 0.3$. Then $min(control(s) = 0.3, rob(s) = 0.4) = 0.3 = min(control(s') = 0.3, rob(s') = 0.3) = 0.3$ and so $s =_{Mixed} s'$. □

Theorem 2 *When BB considers a semantics among Risky, Safe, Risky-Mix or Safe-Mix, overestimations via PastFutureUpperBound, and the pruning conditions of type α, it returns solutions that are optimal according to the chosen semantics.*

Proof: Assume that the best solution found so far in the BB procedure is associated to the pair $\langle lb_a, lb_b\rangle$. Moreover, assume that the partial assignment t is associated to the 4-tuple $\langle a_i(t), a_{ni}(t), b_i(t), b_{ni}(t)\rangle$. Let us denote $uba(t)$ (resp., $ubb(t)$) the value obtained computing $min(a_i(t), a_{ni}(t))$ (resp., $min(b_i(t), b_{ni}(t))$). Assume that S is Risky, Safe, Risky-Mix, or Safe-Mix.

To prove the theorem, we have to show that, if we consider S, and if we prune the subtree below t considering the conditions of type α, then $\langle lb_a, lb_b\rangle$ is, according to S, better than or equal to $\langle uba(t), ubb(t)\rangle$. This means, by definition of S, that one of the following conditions must hold: (i) ($lb_a > uba(t)$) or (ii) ($lb_a = uba(t)$ and $lb_b \geq ubb(t)$). Let us examine each condition of type α in turn:

- If $a_i(t) < lb_a$, then $uba(t) = min(a_i(t), a_{ni}(t)) \leq a_i(t) < lb_a$, thus condition (i) holds.

- If $a_i(t) \geq lb_a$ and $a_{ni}(t) < lb_a$, then $uba(t) = min(a_i(t), a_{ni}(t)) \leq a_{ni}(t) < lb_a$, thus condition (i) holds.

- If $a_i(t) \geq lb_a$, $a_{ni}(t) = lb_a$, and $b_i(t) < lb_b$, then $uba(t) = min(a_i(t), a_{ni}(t)) = lb_a$ and $ubb(t) = min(b_i(t), b_{ni}(t)) \leq b_i(t) < lb_b$, thus condition (ii) holds.

- If $a_i(t) \geq lb_a$, $a_{ni}(t) = lb_a$, $b_i(t) = lb_b$, and $b_{ni}(t) \leq lb_b$, then $uba(t) = min(a_i(t), a_{ni}(t)) = lb_a$, and $ubb(t) = min(b_i(t), b_{ni}(t)) \leq b_{ni}(t) \leq lb_b$, thus condition (ii) holds. □

Theorem 3 *When BB considers the Diplomatic semantics, overestimations via PastFutureUpperBound, and the pruning conditions of type β, it returns solutions that are optimal according to Diplomatic.*

Proof: Consider the same notations used in the first part of the proof of Theorem 2. To prove the statement, we have to show that, if we prune the subtree below t considering the conditions of type β, then $\langle lb_a, lb_b\rangle$ is, according to Diplomatic, better than or equal to $\langle uba(t), ubb(t)\rangle$, that means, by definition of Diplomatic, that the following conditions must hold: $lb_a \geq uba(t)$ and $lb_b \geq ubb(t)$. Let us examine each condition of type β in turn:

- If $a_i(t) \leq lb_a$ and $b_i(t) \leq lb_b$, then $uba(t) = min(a_i(t), a_{ni}(t)) \leq a_i(t) \leq lb_a$ and $ubb(t) = min(b_i(t), b_{ni}(t)) \leq b_i(t) \leq lb_b$.

- If $a_i(t) \leq lb_a$, $b_i(t) \geq lb_b$, and $b_{ni}(t) \leq lb_b$, then $uba(t) = min(a_i(t), a_{ni}(t)) \leq a_i(t) \leq lb_a$ and $ubb(t) = min(b_i(t), b_{ni}(t)) \leq b_{ni}(t) \leq lb_b$.

- $a_i(t) \geq lb_a$, $a_{ni}(t) \leq lb_a$, and $b_i(t) \leq lb_b$, then $uba(t) = min(a_i(t), a_{ni}(t)) \leq a_{ni}(t) \leq lb_a$, and $ubb(t) = min(b_i(t), b_{ni}(t)) \leq b_i(t) \leq lb_b$.

- If $a_i(t) \geq lb_a$, $a_{ni}(t) \leq lb_a$, $b_i(t) \geq lb_b$, and $b_{ni}(t) \leq lb_b$, then $uba(t) = min(a_i(t), a_{ni}(t)) \leq a_{ni}(t) \leq lb_a$, and $ubb(t) = min(b_i(t), b_{ni}(t)) \leq b_{ni}(t) \leq lb_b$. □

References

A. Blinowska, L. D. (1993). Medical application of fuzzy logic: fuzzy patient state in arterial hypertension analysis. In *Proc. SPIE, Applications of Fuzzy Logic Technology*, Vol. 2061. SPIE, The International Society for Optical Engineering.

Amgoud, L., & Prade, H. (2004). Using arguments for making decisions: a possibilistic logic approach. In *Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence (UAI-04)*, pp. 10–17. AUAI Press.

Benferhat, S., Dubois, D., Kaci, S., & Prade, H. (2006). Bipolar possibility theory in preference modeling: representation, fusion and optimal solutions. *Information Fusion*, *7*(1), 135–50.

Benferhat, S., Dubois, D., Prade, H., & Williams, M.-A. (2002). A practical approach to revising prioritized knowledge bases. *Studia Logica*, *70*, 105–130.

Bistarelli, S., Fargier, H., Montanari, U., Rossi, F., Schiex, T., & Verfaillie, G. (1996). Semiring-based CSPs and Valued CSPs: Basic Properties and Comparison. In *Over-Constrained Systems (Selected papers from the Workshop on Over-Constrained Systems at CP'95, reprints and background papers)*, Vol. 1106, pp. 111–150. Springer.

Bistarelli, S., Montanari, U., & Rossi, F. (1995). Constraint solving over semirings. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pp. 624–630. Morgan Kaufmann.

Bistarelli, S., Montanari, U., & Rossi, F. (1997). Semiring-based constraint solving and optimization. *Journal of the ACM*, *44*(2), 201–236.

Bistarelli, S., Pini, M. S., Rossi, F., & Venable, K. B. (2006). Bipolar preference problems. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06)*, pp. 705–706. IOS Press.

Bistarelli, S., Pini, M. S., Rossi, F., & Venable, K. B. (2007a). Bipolar preference problems: framework, properties and solving techniques. In *Recent Advances in Constraints. (Selected papers from 2006 CSCLP Workshop)*, Vol. 4651 of *LNAI*, pp. 78–92. Springer.

Bistarelli, S., Pini, M. S., Rossi, F., & Venable, K. (2007b). Uncertainty in bipolar preference problems. In *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming (CP-07)*, Vol. 4741 of *LNCS*, pp. 782–789. Springer.

Bistarelli, S., Pini, M., Rossi, F., & Venable, K. (2008). From soft constraints to bipolar preferences: modelling framework and solving issues. *Journal of Experimental and Theoretical Artificial Intelligence to appear*.

Borgelt, C., Gebhardt, J., & Kruse, R. (2000). Possibilistic graphical models. In *Computational Intelligence in Data Mining*, Vol. 408 of *CISM Series*. Springer, Berlin.

Bosc, P., & Prade, H. (1997). An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain or imprecise databases. In *Uncertainty Management in Information Systems*, pp. 285–324. Kluwer.

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., & Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, *21*, 135–191.

Boutilier, C., Brafman, R. I., Hoos, H. H., & Poole, D. (1999). Reasoning with conditional ceteris paribus preference statements. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pp. 71–80. Morgan Kaufmann.

Brafman, R. I., & Tennenholtz, M. (1996). On the foundations of qualitative decision theory. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, Vol. 2, pp. 1291–1296. AAAI Press.

Brafman, R. I., & Tennenholtz, M. (1997). On the axiomatization of qualitative decision criteria. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)*. AAAI Press.

Cayrac, D., Dubois, D., Haziza, M., & Prade, H. (1996). Handling uncertainty with possibility theory and fuzzy sets in a satellite fault diagnosis application. In *Uncertainty Management in Information Systems*, Vol. 4, pp. 251–269. IEEE Trans. on Fuzzy Systems.

Dechter, R. (2003). *Constraint processing*. Morgan Kaufmann.

Dominguez, J. A., & Klinko, S. J. (2004). Image analysis via fuzzy reasoning approach : Prototype applications at NASA. In *2004 IEEE international conference on fuzzy systems*. IEEE.

Dubois, D., Fargier, H., & Perny, P. (2002). On the limitations of ordinal approaches to decision making. In *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning (KR-02)*, pp. 133–144. Morgan Kaufmann.

Dubois, D., Fargier, H., & Prade, H. (1995). Fuzzy constraints in job shop scheduling. *Intelligent Manufacturing.*, *6*(4), 215–34.

Dubois, D., Fargier, H., & Prade, H. (1996). Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Appl. Intell.*, *6*(4), 287–309.

Dubois, D., Godo, L., Prade, H., & Zapico, A. (1998). Making decision in a qualitative setting: From decision under uncertainty to case-based reasoning. In *Proceedings of KR '98*, pp. 594–605.

Dubois, D., & Prade, H. (1980). *Fuzzy sets and Systems - Theory and Applications*. Academic Press.

Dubois, D., & Prade, H. (1988). *Possibility theory*. Plenum Press.

Dubois, D., & Prade, H. (1993). Belief revision and updates in numerical formalisms: An overview, with new results for the possibilistic framework.. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pp. 620–625.

Dubois, D., & Prade, H. (1995). Possibility theory as a basis for qualitative decision theory.. In *Proceedings of IJCAI'95*, pp. 1925–1930.

Dubois, D., & Prade, H. (2000). *Fundamentals of Fuzzy Sets - Handbooks of Fuzzy Sets Series*. Kluwer Press.

Fargier, H., Lang, J., Martin-Clouaire, R., & Schiex, T. (1995a). A constraint satisfaction framework for decision under uncertainty. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pp. 167–174. Morgan Kaufmann.

Fargier, H., Lang, J., Martin-Clouaire, R., & Schiex, T. (1995b). A constraint satisfaction framework for decision under uncertainty. In *Proc. of Uncertainty in AI95*. UAI Press.

Fargier, H., Lang, J., & Schiex, T. (1996). Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96)*, Vol. 1, pp. 175–180. AAAI Press.

Fargier, H., & Sabbadin, R. (2003). Qualitative decision under uncertainty: back to expected utility. In *IJCAI'03*, pp. 303–308.

Fishburn, P. C. (1979). *Utility Theory for Decision Making*. Krieger.

Freuder, E., & Wallace, R. (1998). Stable solutions for dynamic constraint satisfaction problems. In *Proceedings of CP'98*, pp. 447–461.

Garcia, L., & Sabbadin, R. (2006). Possibilistic influence diagrams. In *Proceedings of ECAI'06*, pp. 372–376.

Howard, R., & Matheson, J. (1984). Influence diagrams.. *Readings on the Principles and Applications of Decision Analysis*, *2*, 719–762.

Huellermeier, E. (2007). *Case-based Approximate Reasoning*. Springer.

Pini, M. S., & Rossi, F. (2005). Uncertainty in soft constraint problems. In *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP-05) - Doctoral paper*, Vol. 3709 of *LNCS*, p. 865. Springer.

Pini, M. S., Rossi, F., & Venable, K. B. (2005). Possibility theory for reasoning about uncertain soft constraints. In *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-05)*, Vol. 3571 of *LNCS*, pp. 800–811. Springer.

Pralet, C., Schiex, T., & Verfaillie, G. (2006a). Decision with uncertainties, feasibilities, and utilities: Towards a unified algebraic framework. In *Proceedings of ECAI'06*, pp. 427–431.

Pralet, C., Schiex, T., & Verfaillie, G. (2006b). Decomposition of multi-operator queries on semiring-based graphical models. In *Proceedings of CP'06*, pp. 437–452.

Raufaste, E., Neves, R. D. S., & Marin, C. (2003). Testing the descriptive validity of possibility theory in human judgements of uncertainty. *Artificial Intelligence*, *148*, 197–218.

Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of constraint programming*. Elsevier.

Ruttkay, Z. (1994). Fuzzy constraint satisfaction. In *Proceedings 1st IEEE Conference on Evolutionary Computing*, pp. 542–547 Orlando.

Schiex, T. (1992). Possibilistic Constraint Satisfaction problems or ”How to handle soft constraints?”. In *Proceedings of the 8th Annual Conference on Uncertainty in Artificial Intelligence (UAI-92)*, pp. 268–275. Morgan Kaufmann.

Schiex, T., Fargier, H., & Verfaillie, G. (1995). Valued Constraint Satisfaction Problems: Hard and easy problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence, (IJCAI-95)*, pp. 631–639. Morgan Kaufmann.

Tanaka, H., & Guo, P. (1999). *Possibilistic Data Analysis for Operations Research*. Heidelberg: Physica-Verlag.

Walsh, T. (2002). Stochastic constraint programming. In *Proceedings of the 15th Eureopean Conference on Artificial Intelligence (ECAI-02)*, pp. 111–115. IOS Press.

Wolkenhauer, O. (1998). *Possibility Theory with Applications to Data Analysis*. Chichester: Research Studies Press.

Yorke-Smith, N., & Gervet, C. (2003). Certainty closure: A framework for reliable constraint reasoning with uncertainty. In *Proceedings of CP’03*, Vol. 2833 of *LNCS*, pp. 769–783. Springer.

Zadeh, L. (1999). Fuzzy sets as a basis for the theory of possibility. *Fuzzy sets and systems*, *100*, 9–34.