Answer Set Programming for Computing Decisions Under Uncertainty

Roberto Confalonieri[1] and Henri Prade[2]

[1] Universitat Politècnica de Catalunya
Dept. Llenguatges i Sistemes Informàtics
C/ Jordi Girona Salgado 1-3
E - 08034 Barcelona
confalonieri@lsi.upc.edu

[2] Institut de Recherche en Informatique Toulouse (IRIT)
Universitè Paul Sabatier
118 Route de Narbonne
31062 Toulouse Cedex 9, France
prade@irit.fr

Abstract. Possibility theory offers a qualitative framework for modeling decision under uncertainty. In this setting, pessimistic and optimistic decision criteria have been formally justified. The computation by means of possibilistic logic inference of optimal decisions according to such criteria has been proposed. This paper presents an Answer Set Programming (ASP)-based methodology for modeling decision problems and computing optimal decisions in the sense of the possibilistic criteria. This is achieved by applying both a classic and a possibilistic ASP-based methodology in order to handle both a knowledge base pervaded with uncertainty and a prioritized preference base.

1 Introduction

Existing Answer Set Programming (ASP)-based methodologies for handling decision making problems [2,13] amount to compile a decision problem as a logic program able to generate the space of possible decision solutions and to specify an order between them by means of an ordered disjunction connective [4]. Although such approaches are enough to cover decisions in completely certain environment, they become less effective when the knowledge is pervaded with uncertainty. Moreover the existing methods consider empirical decision rules.

The decision under uncertainty problem with qualitative preferences and uncertainty has been studied in the setting of possibility theory assuming a commensurateness hypothesis between the level of certainty and the preferences priority. As in classical utility theory, pessimistic and optimistic criteria have been proposed and justified on the basis of postulates [11]. This approach has been adapted in the setting of possibilistic propositional logic where the available knowledge is described by formulas which are more or less certainly true, and the goals are described in a separate prioritized propositional base.

This paper intends to propose a counterpart of the possibilistic logic-based decision setting within two ASP-based frameworks: Logic Programs with Ordered Disjunction (LPODs) [4] and its possibilistic extension, Logic Programs with Possibilistic Ordered Disjunction (LPPODs) [6]. The motivation behind this work is twofold. First, it is interesting to bridge ASP with qualitative decision making under uncertainty, since to the best of our knowledge any proposal has been made in this respect. Secondly, the use of ASP allows to compute optimal decision in a practical way. Hence, although we do not address implementation issues here, our approach can be implemented on top of two existing ASP-based solvers, *psmodels*[3] and *posPsmodels*[4] which provide a computation of the LPODs and LPPODs semantics.

The paper is structured as follows. After presenting some background concepts about qualitative decision in the possibilistic setting (Section 2), we address the decision problem in ASP by means of LPODs when there is no uncertainty and no priority between the goals (Section 3). Then, we extend this result to the general case with uncertainty and prioritized preferences using LPPODs (Section 4). We compare the proposed approach to previous works in Section 5. Finally, Section 6 concludes the paper.

2 Qualitative Decision in Stratified Propositional Bases

The logical view of a decision problem can be stated in the following way. Let K be the knowledge base describing what is known about the world, D be the set of decision literals, and P another base describing goals delimiting preferred states of the world. Then, a decision, defined as a conjunction d of decision literals such that $K \wedge d \vdash P$ (with $K \wedge d$ consistent) is for sure a good decision (if it exists) since it makes certain that all the goals are satisfied. Looking only for such a decision corresponds to a cautious, pessimistic, attitude. A much more optimistic attitude would correspond to consider also potential decisions d such that $K \wedge d \wedge P \neq \bot$ (which expresses that the possibility of satisfying all the goals remains open).

These two points of view can be extended to the case where K and P are possibilistic logic bases [7], *i.e.* when uncertainty and preferences are matters of degrees. Then K is a set of more or less certain pieces of knowledge and P is a set of goals with associated levels of priority. The certainty and priority levels are supposed to belong to the same linearly ordered scale $\mathcal{S}$ made of $n+1$ levels $\alpha_1 = \mathbf{1} > \alpha_2 > \ldots > \alpha_n > \alpha_{n+1} = \mathbf{0}$. Two sets of postulates for qualitative decision have been proposed that turn to be respectively equivalent to the maximization of a pessimistic criterion and of an optimistic one [11,10]. These two criteria are respectively estimating the necessity and the possibility that a sufficiently satisfactory state is reached (in the sense of qualitative possibility theory). The exact counterpart of these two criteria, when the knowledge and the preferences are expressed under the form of two possibilistic knowledge bases,

[3] `http://www.tcs.hut.fi/Software/smodels/priority/`
[4] `https://github.com/rconfalonieri/posPsmodels`

have been defined in [12,9]. Given n as an order reversing map of scale $\mathcal{S}$ such that $n(\alpha_i) = \alpha_{n+2-i}$ $(1 \leq i \leq n+1)$, K_α as the set of formulas in K having certainty at least equal to α without their certainty levels, and $P_{\underline{\beta}}$ as the set of goals having a priority strictly greater than β without their priority levels, the following criteria are defined [12,9]:

Definition 1. *The pessimistic utility $u_*(d)$ of a decision d is the maximal value of $\alpha \in [0,1]$ such that $K_\alpha \wedge d \vdash P_{\underline{n(\alpha)}}$.*

The optimistic utility $u^*(d)$ of a decision d is the maximal value of $n(\alpha) \in [0,1]$ such that $K_{\underline{\alpha}} \wedge d \wedge P_{\underline{\alpha}} \neq \bot$.

with the convention $\max \emptyset = \mathbf{0}$. The intuition below $u_*(d)$ is that we are interested in finding a decision d (if it exists) such that $K_\alpha \wedge d \vdash P_{\underline{\beta}}$ with α high and β low, *i.e.* such that the decision d together with the most certain part of K entails the satisfaction of the goals, even those with low priority. Taking $\beta = n(\alpha)$ requires that the certainty and priority scales be commensurate. The optimistic utility can be understood in a similar way.

The computation of pessimistic and optimistic decisions has been explored in [12], in the context of possibilistic logic, and later on in [9] by proposing an Assumption Truth Maintenance System (ATMS)-based computation procedure.

An alternative way to compute the pessimistic criteria is to apply possibilistic logic resolution rule. In fact, it has been proved that:

Lemma 1. [7] *Let $K = \{(\phi_i, \alpha_i) \mid 1 \leq i \leq n\}$ be a possibilistic knowledge base, $K_\alpha = \{\phi_i \in K \mid \alpha_i \geq \alpha\}$, (p, β) be a possibilistic formula, and d a literal. Then $K_\alpha \wedge d \vdash_c p$ if and only if $\exists \alpha$ s.t. $K \wedge (d,1) \vdash_p (p, \alpha)$ and $\alpha \geq \beta$, where $\vdash_c$ and $\vdash_p$ are the classical and possibilistic logic inference respectively.*

The aim of this paper is to characterize the qualitative decision making problem under uncertainty in the setting of ASP. Along the paper we will use a running example taking from [12] to exemplify our approach.

Example 1. An agent is supposed i) to know that *if I have an umbrella then I will be not wet*; *if it rains and I do not have an umbrella, then I will be wet*; and *typically if it is cloudy it will rain* (this rule is uncertain) ii) it is known that the sky is cloudy, and iii) being not wet is more important than not carrying an umbrella. The problem then is to decide whether or not to take an umbrella.

3 Making Decision in ASP

In this section we translate a decision problem into a problem tractable by an ASP-based computation. Since the similarity between decision making and abduction is striking [15], it is natural to encode a decision problem by means of LPODs [4] which have been used by Brewka to model abduction [3].

3.1 LPODs and Abduction

In this section we recall the basic notions underlying LPODs [4], and its use in modeling an abduction problem [3]. In the following we assume that the reader has some knowledge about *answer set semantics* (for details see [1]).

Let us consider a propositional language $\mathcal{L}$, with atomic symbols called atoms. A literal is an atom or a negated atom (by classical negation $\neg$). LPODs are sets of rules using ordered disjunction $\times$ in the head of rules to express preferences among literals in the head. An LPOD P is a finite set of ordered disjunction rules of the form $c_1 \times \ldots \times c_k \leftarrow \mathcal{B}^+ \wedge \mathit{not}\ \mathcal{B}^-$, where $\mathcal{B}^+ = \{b_1, \ldots, b_m\}$ and $\mathcal{B}^- = \{b_{m+1}, \ldots, b_{m+n}\}$, and the c_i's $(1 \leq i \leq k)$ and b_j's $(1 \leq j \leq m+n)$ are literals. An ordered disjunction rule (rule for short) r says that if body is satisfied then some c_i must be in the answer set, most preferably c_1, or c_2 if c_1 is impossible, *etc.* If $\forall r \in P$, $k = 1$, then P is an *extended normal program* (*i.e.* $\times$-free); if $k = 1$ and $n = 0$, then P is an *extended definite logic program* (*i.e.* $\times$- and *not*-free). Rules with empty bodies are also known as *facts* (as usual we omit $\leftarrow$) and rules with empty heads are special rules also called *constraints*.

An answer set of an LPOD P is defined as any consistent set of literals M (a and $\neg a$ do not belong to the same set) such that M is a minimal model of the reduced program $P_\times^M$ and that satisfies each rule of P (with $P_\times^M = \bigcup_{r \in P} r_\times^M$ where $r_\times^M = \{c_i \leftarrow \mathcal{B}^+ \mid c_i \in M \wedge M \cap (\{c_1, \ldots, c_{i-1}\} \cup \mathcal{B}^-) = \emptyset\}$). An answer set M can satisfy rules like r to different degrees, where smaller degrees are better. Intuitively, if the body of r is satisfied, then the satisfaction degree is the smallest index i such that $c_i \in M$ (where c_i is in the head of r). Otherwise, the rule is irrelevant and it does not count. Thus, based on the satisfaction degrees of single rules a global preference ordering on answer sets is defined. The comparison criterion between two answer sets M_1 and M_2 is Pareto-based: M_1 is preferred to M_2 ($M_1 \succ M_2$) if and only if there is a rule satisfied better in M_1 than in M_2, and no rule is satisfied better in M_2 than in M_1.

Example 2. Let an LPOD P consist of rules $\{r_1 = a \times b \leftarrow \mathit{not}\ c, r_2 = b \times c \leftarrow \mathit{not}\ d\}$. Then P has three answer sets $M_1 = \{a, b\}$, $M_2 = \{c\}$, $M_3 = \{b\}$ with $M_1 \succ M_2$, $M_1 \succ M_3$, while $M_2 \not\succ M_3$ and $M_3 \not\succ M_2$.

Abduction is the process of generating explanations for a set of observations. An abduction problem usually consists of a set of formulas H of possible explanations, a set of formulas K representing background knowledge, and a set of formulas O describing the observations to be explained. Then, an explanation is a minimal subset H' of H such that $H' \cup K$ is consistent and $H' \cup K \models O$. Brewka [3] has proposed an encoding for the abduction based on LPODs and the credulous inference relation $\models_c$ under answer set semantics.

Definition 2. *Given an LPOD P and a set of literals S, $P \models_c S$ holds, if $\exists M \in SEM_{LPOD}(P)$ such that $S \subseteq M$, where $SEM_{LPOD}(P)$ is the mapping which assigns to P the set of all answer sets of P.*

Example 3. Let P be the LPOD in Example 2. Therefore the following consequences are valid $P \models_c \{a\}$, $P \models_c \{b\}$, $P \models_c \{a, b\}$, $P \models_c \{c\}$.

Algorithm 1 computePessimisticDecisions(DM) : $\langle label_K(Pref), u_* \rangle$

Input: $\{A\ DM = \langle K, D, Pref \rangle$

Output: $\begin{cases} label_K(Pref) : \text{optimal decisions} \\ u_* : \text{pessimistic utility} \end{cases}$

$label_K(Pref) \leftarrow \emptyset$; $u_* \leftarrow \mathbf{0}$;
$P_{dm} \leftarrow decisionMakingToLPOD(DM)$
if $(SEM_{LPOD}(P_{dm}) \neq \emptyset)$ **then**
  $M \leftarrow maxPreferredAS(SEM_{LPOD}(P_{dm}))$
  $label_K(Pref) \leftarrow getDecisionLiterals(D, M)$
  $u_* \leftarrow 1$
end if
return $\langle label_K(Pref), u_* \rangle$

3.2 Fully Certain Knowledge and All-or-nothing Preferences

We propose to translate a decision problem into a problem encoded by an LPOD. In the following we restrict preferences to literals for space reason, even if $\models_c$ could be extended to any propositional formula [5].

Definition 3. *A decision making problem DM is represented as a tuple* $\langle K, D, Pref \rangle$ *where K is an extended definite logic program,* $D = \{d_1, \ldots, d_m\}$ *is a set of decision literals, and* $Pref = \{p_1, \ldots, p_n\}$ *is a set of preference literals.*

Example 4. Let us consider the decision problem in Example 1 without any uncertainty and keeping all the goals as equally important. Then, $K = \{r_1 = \neg w \leftarrow u, r_2 = w \leftarrow r \wedge \neg u, r_3 = r \leftarrow c, r_4 = c\}$, $D = \{u, \neg u\}$, and $Pref = \{\neg w, \neg u\}$.[5]

As in the case of the logical view of the decision problem, we can define optimal decisions according to the pessimistic criteria (the optimistic case could be handled in a similar way). When K expresses completely certain knowledge and $Pref$ are all-or-nothing, optimal decisions, according to a pessimistic point of view, are decisions that, in conjunction with the knowledge, lead to the satisfaction of all the preferences:

Definition 4. *Given a* $DM = \langle K, D, Pref \rangle$, *an optimal pessimistic decision is a minimal set of decision literals* $\Delta \subseteq D$ *such that* $K \cup \Delta \models_c Pref$. *This set is called the label of Pref and it is denoted by* $label_K(Pref)$.

3.3 Computation of Optimal Pessimistic Decisions

The computation of an optimal pessimistic decision is shown in Algorithm 1. The basic DM translation is performed according to Definition 5 where the main construction is borrowed from [3].

[5] We leave $\neg$ negated atoms explicit, although in ASP it is common to replace them with new atoms symbols not belonging to the signature of the program [1].

Definition 5. *Given a* $DM = \langle K, D, Pref \rangle$*, a decision* Δ *for* DM *is computed by an LPOD* $P_{dm}(\langle K, D, Pref \rangle) = P_K \cup \{\leftarrow not\ p \mid p \in Pref\} \cup \{\neg ass(d) \times ass(d) \mid d \in D\} \cup \{d \leftarrow ass(d) \mid d \in D\}$*, where* $ass(d)$ *reads* d *is assumed.*

The generated LPOD P_{dm} can be explained as follows: the use of ordered disjunction rules generates all the possible combinations of decisions, while the use of constraints eliminates the answer sets where preferences are not satisfied. As such, once the answer sets of P_{dm} are computed ($SEM_{LPOD}(P_{dm})$), the optimal set of decisions (`getDecisionLiterals(D,M)`) which are minimal ($label_K(Pref)$) belongs to the most preferred answer set only (`maxPreferredAS` `(`$SEM_{LPOD}(P_{dm})$`)`).

Proposition 1. *Let* $DM = \langle K, D, Pref \rangle$ *be a decision making problem and let* P_{dm} *be the LPOD generated by Algorithm 1. Then,* $\Delta \in label_K(Pref)$ *is an optimal pessimistic decision iff there is a consistent maximally preferred answer set* M *of* P_{dm} *such that* $\Delta = \{d \in D \mid ass(d) \in M\}$*.*

Example 5. Let us consider the DM in Example 4 and its computation with Algorithm 1. LPOD $P_{dm} = \{r_1 = \neg w \leftarrow u, r_2 = w \leftarrow r \wedge \neg u, r_3 = r \leftarrow c, r_4 = c, r_5 = \neg ass(u) \times ass(u), r_6 = \neg ass(\neg u) \times ass(\neg u), r_7 = u \leftarrow ass(u), r_8 = \neg u \leftarrow ass(\neg u), r_9 = \leftarrow not\ \neg w, r_{10} = \leftarrow not\ \neg u\}$. By LPOD semantics, in this case, there is not any answer set which can satisfy all the preferences. As such, the set of best decisions is empty and $u_* = \mathbf{0}$.

Similarly to what happens in possibilistic logic, this criterion can be extended to the case where K is a possibilistic logic program and $Pref$ is a set of possibilistic literals, *i.e.* when uncertainty and preferences are matters of degrees.

4 Making Decision Under Uncertainty in ASP

To be able to capture uncertain knowledge and graded preferences we first introduce LPPODs [6], the possibilistic extension of LPODs.

4.1 Basic Definitions of LPPODs

LPPODs are a recently defined logic programming framework based on LPODs and possibilistic logic [6]. An LPPOD is a finite set of possibilistic ordered disjunction rules of the form $r = \alpha : c_1 \times \ldots \times c_k \leftarrow \mathcal{B}^+ \wedge\ not\ \mathcal{B}^-$, where $\alpha \in \mathcal{S}$ and $c_1 \times \ldots \times c_k \leftarrow \mathcal{B}^+ \wedge\ not\ \mathcal{B}^-$ is an ordered disjunction rule as defined in Section 3.1. $N(r) = \alpha$ is the necessity degree representing the certainty level of the information described by r. A *possibilistic definite program* is defined in a similar way as in Section 3.1. Rules with empty bodies are known as *possibilistic facts* and rules with empty heads are called *possibilistic constraints*.

A *possibilistic literal* is a pair $p = (l, \beta) \in \mathcal{L} \times \mathcal{S}$ where $\mathcal{L}$ is a set of literals and $\mathcal{S}$ a linearly ordered scale. $N(p) = \beta$, while the projection $*$ for a possibilistic literal p is defined as $p^* = l$. Given a set of possibilistic literals M, the projection of $*$ over M is defined as $M^* = \{p^* \mid p \in M\}$. The projection $*$ for a possibilistic ordered disjunction rule r, is $r^* = c_1 \times \ldots \times c_k \leftarrow \mathcal{B}^+ \wedge \textit{not } \mathcal{B}^-$ and the projection of $*$ over P is defined as $P^* = \{r^* \mid r \in P\}$. Notice that P^* is an LPOD.

The LPPODs semantics is defined in terms of a possibilistic counterpart of the program reduction $P_{\times}^{M^*}$ (which reduces an LPPOD to a possibilistic definite program, see [6]) and of a possibilistic consequence operator ΠT_P (which characterizes the possibilistic stable semantics for possibilistic definite programs in terms of a possibilistic minimal model ΠCn, see [14]). Due to lack of space, the complete definitions of $P_{\times}^{M^*}$ and ΠT_P are omitted and we refer to [6,14]. However, it is worthy to point out that the ΠT_P captures the possibilistic *modus ponens* of possibilistic logic [7]. In [6] it is also shown how LPPODs are a proper generalization of LPODs, thus rule satisfaction degrees and the Pareto-based comparison criterion between possibilistic answer sets are properly generalized as well. As such, this criterion can be used to compare possibilistic answer sets.

Definition 6. [6] *Let P be an LPPOD, M be a set of possibilistic literals such that M^* is an answer set of P^*. M is a possibilistic answer set of P if and only if $M = \Pi Cn(P_{\times}^{M^*})$. $SEM_{LPPOD}(P)$ is the mapping which assigns to P the set of all possibilistic answer sets of P.*

Example 6. Let an LPPOD P consist of rules $\{r_1 = \mathbf{1} : a \times b \leftarrow c, r_2 = \alpha : c\}$, where $\mathbf{0} < \alpha < \mathbf{1}$. P has two possibilistic answer sets $M_1 = \{(a, \alpha), (c, \alpha)\}$, $M_2 = \{(b, \alpha), (c, \alpha)\}$ with $M_1 \succ M_2$.

Based on the above definitions we generalize the notion of $\models_c$ to deal with sets of possibilistic literals as:

Definition 7. *Given an LPPOD P and a set of possibilistic literals S, $P \models_p S$ holds, if $\exists M \in SEM_{LPPOD}(P)$ such that $S \sqsubseteq M$ where the relation between sets of possibilistic literals $\sqsubseteq$ is defined as:*
$S \sqsubseteq M \Longleftrightarrow S^* \subseteq M^* \wedge \forall \varphi, \alpha, \beta, (\varphi, \alpha) \in S \wedge (\varphi, \beta) \in M$ *then* $\alpha \leq \beta$.

4.2 Uncertain Knowledge and Prioritized Preferences

Definitions in Section 3.2 are extended in the following way.

Definition 8. *A decision making problem under uncertainty DMU is represented as a tuple $\langle K, D, Pref \rangle$ where, K is a possibilistic definite logic program, D is a set of decision literals, and $Pref = \{((p_1, \beta_1) \ldots, (p_n, \beta_n)\}$ is a set of possibilistic literals, where $\beta_i \in \mathcal{S}$ is the priority of preference p_i.*

Let K_α denote the α-cut of K as $K_\alpha = \{r^* \in K \mid N(r) \geq \alpha\}$, and let $Pref_\beta$ be the β-cut of $Pref$ as $Pref_\beta = \{(p_i, \beta_i)^* \in Pref \mid \beta_i \geq \beta\}$. We also use the notations $K_{\underline{\alpha}}$ and $Pref_{\underline{\beta}}$ (with $\alpha < \mathbf{1}$ and $\beta < \mathbf{1}$, $\mathbf{1}$ being the top element of the scale) for denoting the set of rules and the set of preferences with certainty and priority strictly greater than α and β respectively. In particular $K_{\underline{\mathbf{0}}} = K^*$ and $P_{\underline{\mathbf{0}}} = Pref^*$ ($\mathbf{0}$ being the bottom element of the scale) where K^* and $Pref^*$ denote the set of rules K and the set of preferences $Pref$ without their certainty and priority levels respectively.

Algorithm 2 computePessimisticDecisionsUU(DMU) : $\langle \mathcal{D}, u_* \rangle$

Input: $\begin{cases} A\ DMU = \langle K, D, Pref \rangle \end{cases}$

Output: $\begin{cases} \mathcal{D} : \text{the set of best pessimistic decisions} \\ u_* : \text{the utility of the best pessimistic decisions} \end{cases}$

$\alpha, u_* \leftarrow \mathbf{0}$; $\mathcal{D} \leftarrow \emptyset$; $finish \leftarrow false$;
while ($not\ finish$) **do**
  $\alpha \leftarrow Inc(\alpha) Inc(\alpha)$ increases value of α into the immediately above value
  if ($\alpha = \mathbf{1}$) **then**
    $finish \leftarrow true$
  end if
  $label_{K_\alpha}(Pref_{\underline{n(\alpha)}}) \leftarrow computePessimisticDecisions(\langle K_\alpha, D, Pref_{\underline{n(\alpha)}} \rangle)$
  if ($label_{K_\alpha}(Pref_{\underline{n(\alpha)}}) = \emptyset$) **then**
    $finish \leftarrow true$
  else
    $u_* \leftarrow \alpha$
    $\mathcal{D} \leftarrow label_{K_\alpha}(Pref_{\underline{n(\alpha)}})$
  end if
end while
return $\langle \mathcal{D}, u_* \rangle$

Example 7. Let us consider the decision problem in Example 1 with uncertainty levels and prioritized goals. Then, $K = \{r_1 = \mathbf{1} : \neg w \leftarrow u, r_2 = \mathbf{1} : w \leftarrow r \wedge \neg u, r_3 = \lambda : r \leftarrow c, r_4 = \mathbf{1} : c\}$, $D = \{u, \neg u\}$, and $Pref = \{(\neg w, \mathbf{1}), (\neg u, \delta)\}$, where $\mathbf{0} < \lambda < \mathbf{1}$ and $\mathbf{0} < \delta < \mathbf{1}$.

Definition 9. *Given a* $DMU = \langle K, D, Pref \rangle$*, an optimal pessimistic decision is a set of decision literals* $\Delta \subseteq \mathcal{D}$ *that maximizes* α *such that* $K_\alpha \cup \Delta \models_c Pref_{\underline{n(\alpha)}}$*. This set is called the label of* $Pref_{\underline{n(\alpha)}}$ *and it is denoted by* $label_{K_\alpha}(Pref_{\underline{n(\alpha)}})$*.*

The above definition expresses the fact that an optimal pessimistic decision belongs to an answer set computed with the most certain part of K and that selected preferences, even those with low priority, are satisfied.

4.3 Classical ASP-based Computation

We are now able to describe an algorithm for the pessimistic case. The algorithm is based on successive computations of labels of formulas of $Pref$ which can be computed on the basis of Algorithm 1.

The behavior of Algorithm 2 can be described as follows. First, only the entire knowledge base and highest labelled preferences are considered. If such label is not empty, then we increase our expectations trying to prove less preferred preferences by means of less knowledge. The procedure stops when a set of preferences cannot be proved.

Example 8. As seen in Example 7, K and $Pref$ contain two layers (both scales are commensurate). First of all, according to function $Inc(\alpha)$, α is incremented to the lowest non-nul value, *i.e.* $\alpha = \min\{\lambda, n(\delta)\}$. Whatever the relative positions of λ and δ, $K_\alpha = K^*$. However, we have the following cases: (i) if $\lambda > n(\delta)$ then $\alpha = n(\delta)$ and we have to compute $label_{K^*}(Pref_{\underline{\delta}})$. This means that $Pref_{\underline{\delta}} = \{\neg w\}$, and Algorithm 1 will return the decision $\{u\}$ as label for this preference. As a next step, $\alpha = \lambda$, but the computation of $label_{K^*}(Pref^*)$ is found to be empty. Therefore the set of best pessimistic decisions is in this case $\mathcal{D} = \{u\}$ with utility $u_* = n(\delta)$. (ii) If $\lambda < n(\delta)$ then $\alpha = \lambda$ and we have to compute $label_{K^*}(Pref_{\underline{n(\lambda)}})$. As $n(\lambda) > \delta$, $Pref_{\underline{n(\lambda)}} = \{\neg w\}$, and $label_{K^*}(\neg w) = \{u\}$. A next step is performed where $\alpha = n(\delta)$ and $label_{K_{n(\delta)}}(Pref_{\underline{\delta}}) = \{u\}$. We then have to perform a last step, where $\alpha = \mathbf{1}$, but the computation of $label_{K_{\mathbf{1}}}(Pref^*)$ is equal to $\emptyset$. Therefore, the set of optimal decisions is in this case $\mathcal{D} = \{u\}$ with utility $u_* = n(\delta)$. (iii) If $\lambda = n(\delta)$ then $\alpha = \lambda = n(\delta)$ and we have to compute $label_{K^*}(Pref_{\underline{\delta}})$ which is equal to the computation of $label_{K^*}(\neg w)$ which returns $\{u\}$. Then a next step is performed where $\alpha = \mathbf{1}$ but $label_{K_{\mathbf{1}}}(Pref^*) = \emptyset$.

Thus the best pessimistic solution of the running example is always to take an umbrella with utility $n(\delta)$. Notice that here optimal pessimistic decisions does not depend on the exact value of λ and δ, and even not on their relative positions. However, in the general case, only the positions of the priority and certainty levels matter.

Proposition 2. *Let $DMU = \langle K, D, Pref \rangle$ be a decision making problem under uncertainty and let P_{dm} be the LPOD generated by Algorithm 1. Then, $\Delta \in label_{K_\alpha}(Pref_{\underline{n(\alpha)}})$ such that $\Delta \subseteq D$ is an optimal pessimistic decision maximizing α iff there is a consistent maximally preferred answer set M of P_{dm} such that $\Delta = \{d \in D \mid ass(d) \in M\}$.*

4.4 Possibilistic ASP-based Computation

In the previous section we have provided a method for computing pessimistic decisions reducing the problem to a successive computation of preference labels. In general, the computation of pessimistic decisions (and pessimistic utility) can also be realized by means of an approach closer to possibilistic logic inference, based on the LPPODs semantics. This view is motivated by the possibilistic logic property expressed in Lemma 1.

Algorithm 3 describes an LPPOD-based procedure to compute the set of pessimistic decisions. P_{dm} is constructed by a method `decisionMakingToLPPOD(DMU)` which generalizes Definition 5. To each rule of P_{dm} it associates the corresponding necessity values. However, preference constraints are not added, since the preference satisfaction is checked by means of $\models_p$.

Example 9. Given the DMU in Example 7, the corresponding LPPOD P_{dm} is $\{r_1 = \mathbf{1} : \neg w \leftarrow u, r_2 = \mathbf{1} : w \leftarrow r, \neg u, r_3 = \lambda : r \leftarrow c, r_4 = \mathbf{1} : c, r_5 = \mathbf{1} : \neg ass(u) \times ass(u), r_6 = \mathbf{1} : \neg ass(\neg u) \times ass(\neg u), r_7 = \mathbf{1} : u \leftarrow ass(u), r_8 = \mathbf{1} : \neg u \leftarrow ass(\neg u)\}$.

Algorithm 3 computePessimisticDecisionsUU(DMU) : $\langle \mathcal{D}, u_* \rangle$

Input: $\{A\ DMU = \langle K, D, Pref \rangle$

Output: $\begin{cases} \mathcal{D} : \text{the set of best pessimistic decisions} \\ u_* : \text{the utility of the best pessimistic decisions} \end{cases}$

```
D ← ∅; finish ← false
γ* ← 1; β ← 1; hashTable.push(1, 1);
P_dm ← decisionMakingToLPPOD(DMU)
while (γ* ≥ n(β)) and (not finish) do
  if (P_dm ⊨_p (p_β, β_β)) then
    D ← getDecisionLiterals(SEM_LPPOD(P_dm), p_β)
    γ_β ← getNecessityValue(SEM_LPPOD(P_dm), p_β)
    Dec(β)
    hashTable.push(γ_β, β^+)
    γ* ← γ_β
  else
    finish ← true
  end if
end while
γ* ← hashTable.getMin(key)
β ← hashTable.getValue(γ*)
return ⟨D, min{γ*, n(β)}⟩
```

Returning to the description of the algorithm, γ^* is the certainty value according to which preferences belonging to a stratum $n(\beta)$ of the preference base have been satisfied. We store such information in a $hashTable$ with tuples of the form $\langle Necessity, Priority \rangle$. We start trying to satisfy preferences at the highest strata, and then we increase our expectation decreasing β by $Dec(\beta)$. According to Definition 7 a preference literal is satisfied if and only if its certainty value in a maximally preferred possibilistic answer set of P_{dm} is greater than its priority. If the preference can be proved, the set of decisions satisfying the preferences, its certainty level and the stratum are saved. We keep on iterating until $\gamma^* \geq n(\beta)$ or until we are finished. When a preference belonging to a stratum β is missed, all preferences from stratum β_1 to stratum just before β, *i.e.* β^+ have been satisfied and one of them has the weakest proof γ^*. Then the pessimistic utility corresponds to the minimum value between the γ^* and $n(\beta)$ where β is the stratum to which the preference proved with γ^* belongs.

Example 10. At the beginning $\gamma^* = \mathbf{1}$ and $\beta = \mathbf{1}$, *i.e.* we try to satisfy higher prioritized preferences with the most certain part of K. By applying the LPPOD semantics to P_{dm} in Example 9 two maximally preferred possibilistic answer sets are obtained: $M_1 = \{(\neg u, 1), (ass(\neg u), 1), (c, 1), (r, \lambda), (w, \lambda), (\neg ass(u), 1)\}$ and $M_2 = \{(u, 1), (ass(u), 1), (c, 1), (r, \lambda), (\neg w, 1), (\neg ass(\neg u), 1)\}$. $P_{dm} \models_p (\neg w, 1)$ since $(\neg w, 1) \sqsubset M_2$. Thus, $\mathcal{D} = M_2^* \cap D = \{u\}$ and $\gamma^* = Nec(\neg w) = 1$. While the next level of β to be considered is δ we are sure that at least preference to stratum higher than δ, *i.e.* δ^+, can be satisfied with certainty $\gamma_1 = 1$. Thus

$\gamma^* = \gamma_1$. Since $\gamma^* \geq n(\delta)$ whatever δ value is, we try to satisfy $(\neg u, \delta)$. It is easy to see how $(\neg u, \delta) \not\sqsubseteq M_1$ and $(\neg u, \delta) \not\sqsubseteq M_2$. Thus, we are finished. Then the set of best pessimistic decisions is $\{u\}$ with an utility $u_* = \min\{1, n(\delta^+)\}$, *i.e.* $u_* = n(\delta^+)$. This agrees, as expected, with the label-based computation presented in the previous section.

Proposition 3. *Let* $DMU = \langle K, D, Pref \rangle$ *be a decision making problem under uncertainty, let* P_{dm} *be the LPPOD built using Algorithm 3. Then,* $\Delta \in label_{K_\alpha}(Pref_{\underline{n(\alpha)}})$ *s.t.* $\Delta \subseteq D$ *is an optimal pessimistic decision maximizing* α *iff there exists a consistent possibilistic answer set* M *of* $P_{dm_\alpha} = \{r \in P_{dm} \mid N(r) \geq \alpha\}$ *s.t.* $\{(p_i, \beta_i) \in Pref \mid \beta_i > n(\alpha)\} \sqsubseteq M$ *and* $\Delta = \{d \in D \mid ass(d) \in M^*\}$.

5 Related Work

To the best of our knowledge there are only few works in the literature about modeling qualitative decision problems in ASP [2,13]. These two approaches use the ordered disjunction connective $\times$ introduced in [4] to represent preferences to rank-order different possible states of the world represented as different answer sets. However, they differ on the way uncertainty is handled. In [2] uncertainty is not explicitly represented, since the method is based on the assumption that states of the world which are not normal are disregarded, while taken-into-account states are considered plausible. As such, states can be either negligible or plausible. But, in the latter case, no distinction between the degrees of plausibility of the states can be made, and no further distinctions between the generated answer sets are possible. Grabos in [13] proposed to use $\times$ not only for modeling preferences but also for modeling the plausibility degrees of states. Depending whether a commensurability assumption between the two degrees of plausibility and of preferences is made (or not), decision rules give more importance (or not) to one of the degree in order to select the best answer set according to the attitude of the decision maker w.r.t. the risk. Although this method offers a way to represent uncertainty, decision rules are empirical and are not based on postulates like the possibilistic criteria. Moreover, although our commensurability assumption is a strong assumption, it has been noticed in [8] that working without it leads to an ineffective decision method.

6 Concluding Remarks

In this paper we have presented an ASP-based methodology to compute decision making problems under uncertainty by considering two knowledge bases whose degrees of certainty and priority are commensurate. We have first shown how to encode fully certain knowledge and all-or-nothing preferences, and then, on top of that, how to compute optimal pessimistic decisions.

The reader may be concerned why we have chosen not to take into account ASP optimization techniques (via objective functions) and to compute preferences at meta-level rather than inside LPODs and LPPODs. Our design choice can be motivated by the need of handling two separate knowledge bases and of having a formal handling of uncertainty (in terms of possibilistic logic). In this way we have been able to provide a possibilistic ASP-based methodology which computes the same decisions of the label-based computation. This result agrees both with the classical and the possibilistic resolution views for computing optimal decisions in possibilistic logic.

As general improvements, the decision method used is not able to identify decisions that may satisfy all the goals from the highest level to the lowest one, except one goal at some level β. In fact the algorithm stops at the first unsatisfied preference and does not proceed with preferences at lower strata. The algorithm can be modified accordingly to deal with this case. We also plan to extend the definition of $\models_c$ and $\models_p$ to handle more complex preferences expressions.

References

1. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
2. Brewka, G.: Answer Sets and Qualitative Decision Making. Synthese 146, 171–187 (2005)
3. Brewka, G.: Answer Sets and Qualitative Optimization. Logic Journal of the IGPL 14(3), 413–433 (2006)
4. Brewka, G., Niemelä, I., Syrjänen, T.: Logic Programs with Ordered Disjunction. Computational Intelligence 20(2), 333–357 (2004)
5. Brewka, G., Niemelä, I., Truszczynski, M.: Answer Set Optimization. In: Gottlob, G., Walsh, T. (eds.) IJCAI03. pp. 867–872. Morgan Kaufmann (2003)
6. Confalonieri, R., Nieves, J.C., Osorio, M., Vázquez-Salceda, J.: Possibilistic Semantics for Logic Programs with Ordered Disjunction. In: Link, S., Prade, H. (eds.) Foundations of Information and Knowledge Systems, FoIKS 2010, Proceedings. LNCS, vol. 5956. Springer-Verlag, Berlin (2010)
7. Dubois, D., Lang, J., Prade, H.: Possibilistic logic, In Handbook of logic in artificial intelligence and logic programming, pp. 439–513. Oxford Univ. Press, NY, (1994)
8. Dubois, D., Fargier, H., Prade, H., Perny, P.: Qualitative decision theory: from Savage's axioms to nonmonotonic reasoning. J. ACM 49(4), 455–495 (2002)
9. Dubois, D., Le Berre, D., Prade, H., Sabbadin, R.: Using Possibilistic Logic for Modeling Qualitative Decision: ATMS-based Algorithms. F. Inf. 37, 1–30 (1999)
10. Dubois, D., Prade, H.: Possibility theory as a basis for qualitative decision theory. In: IJCAI95. pp. 1924–1930. Morgan Kaufmann, San Francisco, USA (1995)
11. Dubois, D., Prade, H., Sabbadin, R.: Decision-theoretic foundations of qualitative possibility theory. Europ. Journal of Operational Research 128(3), 459–478 (2001)
12. Dubois, D., Prade, H., Sabbadin, R.: A Possibilistic Logic Machinery for Qualitative Decision. In: In AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning. pp. 47–54. AAAI Press (1997)
13. Grabos, R.: Qualitative model of decision making. In: Bussler, C., Fensel, D. (eds.) Artificial Intelligence: Methodology, Systems, and Applications, LNCS, vol. 3192, pp. 480–489. Springer Berlin / Heidelberg (2004)
14. Nicolas, P., Garcia, L., Stéphan, I., Lefèvre, C.: Possibilistic uncertainty handling for answer set programming. Annals of Mathematics and Artificial Intelligence 47(1-2), 139–181 (2006)
15. Sabbadin, R.: Decision As Abduction? In: ECAI, pp. 600–604. Wiley (1998)