# A Novel Framework for Reasoning over Optimization Problems in Probabilistic Answer Set Programming

**Damiano Azzolini**[1] , **Giuseppe Mazzotta**[2] , **Francesco Ricca**[2] , **Fabrizio Riguzzi**[1]
[1]University of Ferrara, Ferrara, Italy
[2]University of Calabria, Rende, Italy
{damiano.azzolini, fabrizio.riguzzi}@unife.it, {giuseppe.mazzotta,francesco.ricca}@unical.it

### Abstract

Probabilistic logic-based languages offer an expressive framework for encoding uncertain information in a human-interpretable way. Among existing formalisms, Probabilistic Answer Set Programming (PASP) stands out for its ease of modeling complex scenarios. The current definition of PASP is limited to programs consisting of disjunctive rules and probabilistic facts only. To enhance the expressivity of the framework, we introduce Optimal Probabilistic Answer Set Programming, which extends the language by allowing the inclusion of weak constraints within PASP specifications. We motivate this extension through some real-world application scenarios and present a detailed computational complexity analysis for both the inference and Most Probable Explanation (MPE) tasks.

## 1 Introduction

Statistical Relational Artificial Intelligence (Raedt et al. 2016) is a research field that aims to combine the expressivity of logic-based languages and the flexibility of probability theory and graphical models (Koller and Friedman 2009). A probabilistic logic program (PLP) (Riguzzi 2022) is a logic program extended with constructs, such as *probabilistic facts* (De Raedt, Kimmig, and Toivonen 2007), to represent uncertain data. In this case, the Distribution Semantics (DS) (Sato 1995) assigns a meaning to such programs, and it is adopted in tools such as PRISM (Sato 1995), ProbLog (De Raedt, Kimmig, and Toivonen 2007), and cplint (Riguzzi and Swift 2011). The DS requires that the underlying logic program has a 2-valued well-founded model (Van Gelder, Ross, and Schlipf 1991). Among logic-based formalisms, also Answer Set Programming (Brewka, Eiter, and Truszczyński 2011) (ASP) has been extended to handle probabilistic reasoning leading to the introduction Probabilistic ASP (PASP) (Cozman and Mauá 2020). ASP is a well-known formalism adopted in many real-world applications (Erdem, Gelfond, and Leone 2016; Falkner et al. 2018) thanks to its expressivity (Eiter and Gottlob 1995b; Dantsin et al. 2001) and the availability of efficient and well-maintained solvers (Gebser et al. 2019; Alviano et al. 2017). However, the DS semantics cannot be applied to ASP extended with, for example, probabilistic facts, since answer set programs may be associated with multiple models. To overcome this, the credal semantics (CS) was proposed (Cozman and Mauá 2020). Currently, the CS focuses on programs that do not contain *weak constraints*, which makes it not handy to handle optimization criteria.

In this paper, we extend the CS with weak constraints (Buccafurri, Leone, and Rullo 2000), by introducing the *optimal probabilistic answer set programming* framework. We motivate this extension by showcasing how some real-world problems involving optimization and probability that can be naturally modeled with our formalism. Also, we conduct a theoretical study on the computational complexity of inference and Most Probable Explanation tasks.

The paper is structured as follows: Section 2 discusses the needed background, Section 3 proposes the optimal probabilistic answer set programming framework, Section 4 encodes real-world examples with our formalism, Section 5 provides complexity results, Section 6 surveys related works, and Section 7 concludes the paper.

## 2 Background

**Computational Complexity.** We introduce the relevant counting complexity classes (Valiant 1979; Hemaspaandra and Vollmer 1995), as well as the polynomial time hierarchy (PH) (Stockmeyer 1976). Counting problems are aimed at computing the number of solutions of an instance of a given problem. In general, they fall into the complexity classes introduced by Valiant (1979) and Hemaspaandra and Vollmer (1995). Intuitively, for problems that can be solved by a polynomial-time non-deterministic Turing machine (e.g., $NP$-complete), the function counting the number of accepting paths of NP machines falls into the class denoted by $\#P$. Analogously, for an arbitrary complexity class $C$, $\#C$ denotes the set of functions that count the number accepting paths of a polynomial-time non-deterministic Turing machine with an oracle in the class $C$. However, Hemaspaandra and Vollmer provided an alternative predicate-based definition.

**Definition 1** (Hemaspaandra and Vollmer 1995)**.** *For a complexity class $\mathcal{K}$, $\# \cdot \mathcal{K}$ denotes the set of functions such that for some $\mathcal{K}$-computable binary predicate $R$ and a polynomial $p$ it holds that for every input string $x$:*

$$f(x) = |\{y \mid p(|x|) = |y| \wedge R(x, y)\}|$$

That is, each function $f \in \# \cdot \mathcal{K}$ counts the strings of polynomial length, denoted by $y$, w.r.t. the input size $|x|$, such that the predicate $R(x, y)$ holds.

Between the machine-based definition by Valiant (1979) and the predicated-based definition by Hemaspaandra and Vollmer (1995) there exits a strong relationship: as pointed out by Hemaspaandra and Vollmer (1995), $\#NP = \#coNP$ whereas $\# \cdot NP = \# \cdot coNP$ if and only if $NP = coNP$. Moreover, it holds that $\# \cdot NP \subseteq \#NP = \# \cdot P^{NP} = \# \cdot coNP$ (Durand, Hermann, and Kolaitis 2000). This will be instrumental in our complexity study.

Definition 1 has been tailored also to decision problems, in which we are interested in verifying the lower/upper bounds of the number of solutions.

**Definition 2** (Hemaspaandra and Vollmer 1995)**.** *Let $\mathcal{K}$ be a complexity class and $A$ a problem, then $A \in C \cdot \mathcal{K}$ if there exists a function $f \in \# \cdot \mathcal{K}$ and a polynomial-computable function $g$ such that $x \in A \leftrightarrow f(x) \geq g(x)$.*

That is, a problem $A$ belongs to $C \cdot \mathcal{K}$ if, for every input $x$, the number of solutions for $x$ can be computed by a function $f \in \# \cdot \mathcal{K}$ and $x \in A$ iff $f(x) \geq g(x)$, where $g$ is a polynomial-computable function.

The classes $\Delta_k^P$, $\Sigma_k^P$, and $\Pi_k^P$ of the polynomial hierarchy (PH) (Stockmeyer 1976) are defined as (cf. (Garey and Johnson 1979)): $\Delta_0^P = \Sigma_0^P = \Pi_0^P = P$ and, for all $k > 0$

$$\Delta_{k+1}^P = P^{\Sigma_k^P}, \;\; \Sigma_{k+1}^P = NP^{\Sigma_k^P}, \;\; \Pi_{k+1}^P = coNP^{\Sigma_k^P},$$

where $NP = \Sigma_1^P$, $coNP = \Pi_1^P$, and $P^{NP} = \Delta_2^P$.

In general, $P^C$ (resp. $NP^C$) denotes the class of problems solvable in polynomial time by a deterministic (resp. nondeterministic) Turing machine equipped with an oracle for a class $C$ problem. Using an oracle $O \in C$ to solve a problem means that such a problem can invoke $O$ as a subroutine during its computation, where each oracle call is assumed to take a single computation step. For further details about $NP$-completeness and complexity theory, we refer the reader to the dedicated literature (Papadimitriou 1994).

**Answer Set Programming.** In Answer Set Programming (ASP) (Calimeri et al. 2020) a *term* is either a *variable* (i.e., a string starting with an uppercase letter) or a constant (i.e., a number or a string starting with a lowercase letter). An *atom* is an expression of the form $p(t_1, \ldots, t_n)$, where $p$ is *predicate* of arity $n$ and $t_1, \ldots, t_n$ are terms. A *literal* is an atom $a$ or its negation $not\ a$, where $not$ denotes *negation as failure*. A literal is *ground* if it contains no variables. A literal is *positive* if it is of the form $a$, otherwise it is negative. Given a set of literals $S$, we denote by $S^+$ and $S^-$ the set of positive and negative literals in $S$, respectively. A *rule* is an expression of the form $h_1; \ldots; h_m :- \ l_1, \ldots, l_n$, where $h_1, \ldots, h_m$ is a disjunction of atoms called *head* and $l_1, \ldots, l_n$ is a conjunction of literals referred to as *body*. A *weak constraint* is an expression of the form $:\sim\ l_1, \ldots, l_n\ [w@l, t]$, where $l_1, \ldots, l_n$ are literals referred to as body, $w$ and $l$ are terms referred to as *weight* and *level*, respectively, and $t$ is a (possibly empty) list of terms. If we do not specify the level (i.e., $@l$), then it is assumed to be 0. For a rule $r$, we denote by $H_r$ the set of atoms appearing in the head. For a rule $r$ (resp. a weak constraint $w$) we denote by $B_r$ (resp. $B_w$) the set of literals appearing in the body. A rule $r$ is said to be a *fact* if $|H_r| = 1$ and $B_r = \emptyset$, a *constraint* if $H_r = \emptyset$ and $|B_r| > 0$, or *normal* if $|H_r| = 1$ and $|B_r| > 0$. A rule $r$ is *safe* if each variable appearing in $H_r$ and $B_r^-$ also appears in $B_r^+$. The definition of safety also extends to weak constraints. A weak constraint $w$ is *safe* if each variable appearing in $B_w^-$, $w$, $l$, and $t$ also appears in $B_w^+$. We only consider safe rules and safe weak constraints.

An ASP *program* is a finite set of safe rules and weak constraints. Given a program $\mathcal{T}$, $\mathcal{R}(\mathcal{T})$ and $\mathcal{W}(\mathcal{T})$ denote, respectively, the rules and weak constraints of $\mathcal{T}$. A program $\mathcal{T}$ is said to be *normal* if all the rules in $\mathcal{T}$ are normal and *plain* if $\mathcal{W}(\mathcal{T}) = \emptyset$.

For a program $\mathcal{T}$, we denote by $U_\mathcal{T}$ the set of constants appearing in $\mathcal{T}$ referred to as *Herbrand Universe*, and by $B_\mathcal{T}$ the set of ground atoms that can be obtained by predicates in $\mathcal{T}$ and constants in $U_\mathcal{T}$, referred to as *Herbrand Base*. For a rule $r \in \mathcal{R}(\mathcal{T})$ (resp. a weak constraint $w \in \mathcal{W}(\mathcal{T})$), we denote by $ground(r)$ (resp. $ground(w)$) the set of ground instantiation of $r$ (resp. $w$) obtained by mapping each variable to a constant in $U_\mathcal{T}$. Similarly, we denote by $ground(\mathcal{T})$ the union of ground instantiations for each rule and weak constraint in $\mathcal{T}$.

**Example 1.** *Let $\mathcal{T}$ be the following program:*

```
c(1,2). c(2,1). d(1).
a(X,Y):- d(X), d(Y).
b(Y):- c(X,Y), not a(Y,X).
:~ b(X). [1@X]
```

*Then $ground(\mathcal{T})$ denotes the following ground program*

```
c(1,2). c(2,1). d(1). d(2).
a(1,1):- d(1), d(1).
a(1,2):- d(1), d(2).
a(2,1):- d(2), d(1).
a(2,2):- d(2), d(2).
b(1):- c(1,1), not a(1,1).
b(2):- c(1,2), not a(2,1).
b(1):- c(2,1), not a(1,2).
b(2):- c(2,2), not a(2,2).
:~ b(1). [1@1]
:~ b(2). [1@2]
```

The *dependency graph* of $\mathcal{T}$, denoted by $G_\mathcal{T}$, is a labeled directed graph whose node are atoms in $B_\mathcal{T}$. $G_\mathcal{T}$ includes a positive (resp. negative) edge from $b$ to $h$ if there exists a rule $r \in ground(P)$ such that $b \in B_r^+$ (resp. $not\ b \in B_r^-$) and $h \in H_r$. The *positive dependency graph* of $\mathcal{T}$, denoted by $G_\mathcal{T}^+$, is obtained from $G_\mathcal{T}$ by considering only positive edges. A *component* $C \subseteq B_\mathcal{T}$ is a maximal set of atoms that are strongly connected in $G_\mathcal{T}^+$. A program $\mathcal{T}$ is said to be *stratified* if $G_\mathcal{T}$ does not contain any loop involving negative edges and *head-cycle-free* (HCF) if there is no component $C$ such that there exists a rule $r \in ground(\mathcal{T})$ where $a, b \in H(r)$ and $a, b \in C$. HCF programs can be transformed, in polynomial time, into normal programs by shifting disjunctions (Dix, Gottlob, and Marek 1996).

An *interpretation* $I \subseteq B_\mathcal{T}$ is a set of atoms. A positive (resp. negative) ground literal $l = a$ (resp. $l = not\ a$) is true w.r.t. $I$, denoted by $I \models l$, if $a \in I$ (resp. $a \notin I$), otherwise $l$ is false w.r.t. $I$, denoted by $I \not\models l$. A disjunction of ground literals $l_1; \dots; l_n$ is true w.r.t. $I$ if $I \models l_i$, for some $i \in \{1, \dots, n\}$. A conjunction of ground literals $l_1, \dots, l_n$ is true w.r.t. $I$ if $I \models l_i$, for each $i \in \{1, \dots, n\}$. Let $r \in ground(\mathcal{T})$, then $r$ is satisfied w.r.t. $I$ if $r$ has a false body w.r.t. $I$ or a true head w.r.t. $I$. An interpretation $I$ is a *model* of $\mathcal{T}$ if $I$ satisfies each rule in $ground(\mathcal{T})$.

Let $\mathcal{T}$ be a program and $I$ be an interpretation, then $\mathcal{T}^I$ denotes the *FLP-reduct* (Faber, Pfeifer, and Leone 2011), i.e., the program obtained from $\mathcal{T}$ by removing each rule having a false body w.r.t. $I$. A model $I$ is an *answer set* of $\mathcal{T}$ if $I$ is $\subset$-mininal model of $\mathcal{T}^I$ (i.e., there is no $I' \subset I$ such that $I'$ is a model of $\mathcal{T}^I$). For an ASP $\mathcal{T}$, we denote with $AS(\mathcal{T})$ the set of its answer sets. A program $\mathcal{T}$ with no answer sets is termed *incoherent*. Let $M \in AS(\mathcal{T})$, then weak constraints in $ground(\mathcal{T})$ define the cost of $M$. More precisely, the set of weak constraint violations is defined as $ws(\mathcal{T}, M) = \{(w, l, t) \mid :\sim l_1, \dots, l_n[w@l, t] \in ground(\mathcal{T}), M \models l_1, \dots, l_n\}$. Then, for each integer level $l$, the cost of $M$ is defined as $\mathcal{C}(\mathcal{T}, M, l) = \sum_{(w,l,t) \in ws(\mathcal{T},M)} w$. If there is only one level, we use $\mathcal{C}(\mathcal{T}, M)$ (which is equivalent to $\mathcal{C}(\mathcal{T}, M, 0)$). Let $M_1, M_2 \in AS(\mathcal{T})$, then $M_1$ is *dominated* by $M_2$ if there exists an integer level $l$ such that $\mathcal{C}(\mathcal{T}, M_1, l) > \mathcal{C}(\mathcal{T}, M_2, l)$, and for each $l' > l$, $\mathcal{C}(\mathcal{T}, M_1, l') = \mathcal{C}(\mathcal{T}, M_2, l')$. An answer set $M$ is also an *optimal answer set* for $\mathcal{T}$ if, for each $M' \in AS(\mathcal{T})$, $M$ is not dominated by $M'$. We denote by $AS^*(\mathcal{T})$ the set of optimal answer sets of $\mathcal{T}$.

**Example 2** (Clique.)**.** *The following facts define a graph.*

```
e(1,2). e(1,3). e(1,4). e(2,3). e(3,4).
```

*The problem of finding a clique in the graph can be encoded with the following ASP.*

```
edge_(X,Y):- e(X,Y).
edge_(X,Y):- e(Y,X).
v(X) :- edge_(X,Y).
in(X);out(X) :- v(X).
:- in(X), in(Y), X!=Y, not edge_(X,Y).
```

*This program has 12 answer sets, one per possible clique.*

**Example 3** (Maximal Clique)**.** *If we want to find the maximal clique, we can add to the program of Example 2 the following weak constraint:*

```
:~ out(X). [1@1,X]
```

*This requires finding the answer sets where the number of $out/1$ atoms is minimized. There are two solutions (we only report the $in/1$ facts, for brevity): $\{in(1), in(2), in(3)\}$ and $\{in(1), in(3), in(4)\}$ both of size 3 with cost 1.*

For simplicity we use the ASP acronym in place of ASP program, the context will disambiguate the meaning.

**Probabilistic Answer Set Programming.** We consider a *probabilistic answer set program* (PASP) $\mathcal{P}$ as a plain head-cycle-free ASP $\mathcal{T}$ extended with a set of ground *probabilistic facts* (De Raedt, Kimmig, and Toivonen 2007) $\mathcal{F}$ (i.e., $\mathcal{P}$ is a tuple $(\mathcal{T}, \mathcal{F})$) of the form $\pi :: a$ where $a$ is an atom and $\pi \in [0, 1]$ is its probability. We assume, as usual, that probabilistic facts cannot appear in heads of rules and are independent. A *selection* $\sigma$ is a subset of probabilistic facts and identifies a *world* $w_\sigma$, i.e., an answer set program obtained by adding to $\mathcal{T}$ each atom of a probabilistic fact in $\sigma$. Thus, a PASP with $n$ probabilistic facts has $2^n$ selections and so $2^n$ worlds. We denote with $\mathcal{S}$ and $\Sigma$ the set of selections and worlds, respectively. We consider the *credal semantics* (CS) (Cozman and Mauá 2020). Under the CS, every world is required to have at least one answer set, i.e., to be coherent. The probability of a world $w_\sigma$, $P(w_\sigma)$, is

$$P(w_\sigma) = \prod_{a_i \in w_\sigma} \pi_i \cdot \prod_{a_i \notin w_\sigma} (1 - \pi_i). \tag{1}$$

With a slight abuse of notation, when we write $P(\sigma)$ for a selection $\sigma$ we denote the probability of the corresponding world $w_\sigma$, $P(w_\sigma)$. The probability of a ground atom $q$ (we consider a ground atom without loss of generality since we can always insert in the program a rule with $q$ in the head and a conjunction of ground literals in the body) is described by a lower $\underline{\mathrm{P}}(q)$ and an upper $\overline{\mathrm{P}}(q)$ bound, where

$$\begin{aligned} \underline{\mathrm{P}}(q) &= \sum_{w_\sigma \in \Sigma \mid \forall M \in AS(w_\sigma),\ M \models q} P(w_\sigma), \\ \overline{\mathrm{P}}(q) &= \sum_{w_\sigma \in \Sigma \mid \exists M \in AS(w_\sigma),\ M \models q} P(w_\sigma). \end{aligned} \tag{2}$$

A related task is computing the *most probable explanation* (MPE): given a PASP $\mathcal{P}$ and a ground atom $e$ (called evidence), the most probable explanation for $e$ is the selection $\sigma$ associated with the world with the highest probability (computed with Equation (1)) in which $e$ is cautiously (or bravely) true. That is,

$$\begin{aligned} \underline{\mathrm{MPE}}(e) &= \operatorname*{arg\,max}_{\sigma \in \mathcal{S} \mid \forall M \in AS(w_\sigma), M \models e} P(w_\sigma), \\ \overline{\mathrm{MPE}}(e) &= \operatorname*{arg\,max}_{\sigma \in \mathcal{S} \mid \exists M \in AS(w_\sigma), M \models e} P(w_\sigma). \end{aligned} \tag{3}$$

We denote with $P(\underline{\mathrm{MPE}}(e))$ (resp. $P(\overline{\mathrm{MPE}}(e))$) the probability of the world associated with $\underline{\mathrm{MPE}}(e)$ (resp. $\overline{\mathrm{MPE}}(e)$).

**Example 4** (Clique in Probabilistic Graph)**.** *Consider again Example 2 and suppose that the $e/2$ facts are now probabilistic, i.e., there is uncertainty between the connections:*

```
0.1::e(1,2). 0.2::e(1,3). 0.3::e(1,4).
0.4::e(2,3). 0.5::e(3,4).
```

*The program has 32 worlds. For example, the world where all the $e/2$ facts are included coincides with the program of Example 2 and has probability* $0.1 \cdot 0.2 \cdot 0.3 \cdot 0.4 \cdot 0.5 = 0.0012$. *We can now ask, for example, for the probability of node 1 to be included in a clique (i.e., query the atom $in(1)$), obtaining $\underline{\mathrm{P}}(in(1)) = 0$ and $\overline{\mathrm{P}}(in(1)) = 0.496$, or for the probability that there exists a clique of size 3, by adding $qr :- in(A), in(B), in(C), A! = B, B! = C, A! = C$ and asking $qr$, obtaining $\underline{\mathrm{P}}(qr) = 0$ and $\overline{\mathrm{P}}(qr) = 0.0368$. For both cases, the lower probability is zero: this means that there are no worlds where every answer set entails the query. However, this is not true in general. The upper MPE state for the same atom $qr$ is $\overline{\mathrm{MPE}}(qr) = \{e(1,3), e(1,4), e(3,4)\}$ with probability $P(\overline{\mathrm{MPE}}(qr)) = 0.0162$.*

**Quantified Answer Set Programming.** Here, we introduce Quantified Answer Set Programming (ASP(Q)) which will be instrumental in our complexity studies. ASP(Q) extends ASP by introducing the concept of *quantifiers* over (optimal) answer sets of ASP programs (Amendola, Ricca, and Truszczynski 2019; Mazzotta, Ricca, and Truszczynski 2024). The usage of quantifiers allows modeling problems throughout the entire PH as natural as modeling NP problems in ASP. An ASP(Q) program $\Pi$ is of the form:

$$\Box_1 \mathcal{T}_1 \ldots \Box_n \mathcal{T}_n : C, \tag{4}$$

where for each $i \in \{1, \ldots, n\}$, $\mathcal{T}_i$ is a normal ASP program, $\Box_i \in \{\forall^{st}, \exists^{st}\}$, and $C$ is a normal program both plain and stratified. An ASP(Q) program is said to be *existential* if $\Box_1 = \exists$, otherwise it is *universal*. The ASP(Q) semantics is inductively defined as:

- $\exists^{st} \mathcal{T} : C$ is coherent iff there exists $M \in AS^*(\mathcal{T})$ such that $C \cup fix_{\mathcal{T}}(M)$ is coherent.
- $\forall^{st} \mathcal{T} : C$ is coherent iff for each $M \in AS^*(\mathcal{T})$, $C \cup fix_{\mathcal{T}}(M)$ is coherent.
- $\exists^{st} \mathcal{T}\, \Pi$ is coherent iff there exists $M \in AS^*(\mathcal{T})$ such that $\Pi_{\mathcal{T},M}$ is coherent.
- $\forall^{st} \mathcal{T}\, \Pi$ is coherent iff for each $M \in AS^*(\mathcal{T})$, $\Pi_{\mathcal{T},M}$ is coherent.

where $fix_{\mathcal{T}}(M) = \{a \mid a \in M\} \cup \{ \text{:-}\ a \mid a \in B_{\mathcal{T}} \setminus M\}$, $\Pi$ is of the form (4), and $\Pi_{\mathcal{T},M} = \Box_1 \mathcal{T}_1 \cup fix_{\mathcal{T}}(M) \ldots \Box_n P_n : C$. For an existential ASP(Q) program $\Pi$ of the form (4), $M \in AS^*(\mathcal{T}_1)$ is a *quantified answer set* of $\Pi$ iff $\Box_2 \mathcal{T}_2 \cup fix_{\mathcal{T}_1}(M_1) \ldots \Box_n \mathcal{T}_n : C$ is coherent. $QAS(\Pi)$ denotes the quantified answer sets of $\Pi$.

## 3 Optimal Probabilistic ASP

Let us now introduce the *Optimal* Probabilistic Answer Set Programming framework.

**Definition 3.** *An optimal PASP $\mathcal{P}^{\sim}$ is a tuple $(\mathcal{T}, \mathcal{F}, \mathcal{W})$, where $\mathcal{T}$ is a plain head-cycle-free ASP, $\mathcal{F}$ is a set of probabilistic facts, and $\mathcal{W}$ is a set of weak constraints.*

That is, an *optimal* PASP is a PASP extended with a set of weak constraints. Each world $w_\sigma$ is now equipped with a set of weak constraints imposing optimality criteria over the possible answer sets. Thus, each $w_\sigma$ may have zero (in case it is incoherent) or more *optimal* answer sets. As for PASP without weak constraints (which we call *plain* PASP), we require that each $w_\sigma$ is satisfiable, i.e., has at least one (optimal) answer set. Clearly, if $w_\sigma$ has only one answer set $A$, then $A$ will also be the optimal answer set.

The probability of a query $q$ in an optimal PASP is computed similarly to Equation (2). The only difference is that the summation ranges over the *optimal* answer set. That is, given an optimal PASP $\mathcal{P}^{\sim}$, the lower (resp. upper) probability $\underline{\mathrm{P}}^*(q)$ (resp. $\overline{\mathrm{P}}^*(q)$) for a query $q$ are

$$\begin{aligned} \underline{\mathrm{P}}^*(q) &= \sum_{w_\sigma \in \Sigma \mid \forall M \in AS^*(w_\sigma),\ M \models q} P(w_\sigma), \\ \overline{\mathrm{P}}^*(q) &= \sum_{w_\sigma \in \Sigma \mid \exists M \in AS^*(w_\sigma),\ M \models q} P(w_\sigma). \end{aligned} \tag{5}$$

Similary, MPE for an evidence $e$ is now computed by considering only optimal answer sets, i.e.,

$$\begin{aligned} \underline{\mathrm{MPE}}^*(e) &= \operatorname*{arg\,max}_{\sigma \in \mathcal{S} \mid \forall M \in AS^*(w_\sigma), M \models e} P(w_\sigma), \\ \overline{\mathrm{MPE}}^*(e) &= \operatorname*{arg\,max}_{\sigma \in \mathcal{S} \mid \exists M \in AS^*(w_\sigma), M \models e} P(w_\sigma). \end{aligned} \tag{6}$$

In the following, we will often consider a PASP $\mathcal{P} = (\mathcal{T}, \mathcal{F})$ and its optimal counterpart $\mathcal{P}^{\sim} = (\mathcal{T}, \mathcal{F}, \mathcal{W})$. In this case, for a query $q$, we denote with $\underline{\mathrm{P}}(q)$ and $\overline{\mathrm{P}}(q)$ the probability computed by considering $\mathcal{P}$ and with $\underline{\mathrm{P}}^*(q)$ and $\overline{\mathrm{P}}^*(q)$ the probability computed by considering $\mathcal{P}^{\sim}$. That is, to lighten the notation, we do not insert an explicit dependence on $\mathcal{P}$ nor $\mathcal{P}^{\sim}$.

The next theorem relates the probability of a query computed in a plain PASP to that computed in an optimal PASP.

**Theorem 1.** *Let $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{W}$ be a set of normal rules, probabilistic facts, and weak constraints, respectively, and $q$ an atom (query). Let $\mathcal{P} = (\mathcal{T}, \mathcal{F})$ and $\mathcal{P}^{\sim} = (\mathcal{T}, \mathcal{F}, \mathcal{W})$. Then, $\underline{\mathrm{P}}(q) \leq \underline{\mathrm{P}}^*(q)$ and $\overline{\mathrm{P}}^*(q) \leq \overline{\mathrm{P}}(q)$.*

*Proof.* The number of worlds of $\mathcal{P}$ and $\mathcal{P}^{\sim}$ is the same, as is the probability for each world, since the set of probabilistic facts is the same. Consider a world $w$ in $\mathcal{P}$ and its counterpart $w^*$ for $\mathcal{P}^{\sim}$. Since $AS^*(w^*) \subseteq AS(w)$, if the condition for the world to contribute to the lower bound is satisfied for $\mathcal{P}$, then it will be satisfied for $\mathcal{P}^{\sim}$ but not vice versa (the condition for $\mathcal{P}$ is stronger). Conversely, if the condition for the world to contribute to the upper bound is satisfied for $\mathcal{P}^{\sim}$, then it will be satisfied for $\mathcal{P}$ but not vice versa (the condition for $\mathcal{P}^{\sim}$ is stronger). $\square$

Theorem 1 shows that the semantics of optimal PASP is credal, i.e., is the set of all probability measures that dominate an infinitely monotone Choquet capacity (Cozman and Mauá 2017). As in the credal semantics for plain PASP, the distribution of the probability mass of a world over the optimal answer set is left open. The only difference is that the set of answer sets is a subset of those for plain PASP. Moreover, the following theorems immediately hold.

**Theorem 2.** *Let $\mathcal{P} = (\mathcal{T}, \mathcal{F})$ with query $q$ and $\mathcal{P}^{\sim} = (\mathcal{T}, \mathcal{F}, \mathcal{W})$ with $\mathcal{W} = \{:\sim q\ [C]\}$ with $C < 0$ or with $\mathcal{W} = \{:\sim not\ q\ [C]\}$ with $C > 0$. Then, $\underline{\mathrm{P}}^*(q) = \overline{\mathrm{P}}^*(q) = \overline{\mathrm{P}}(q)$.*

**Theorem 3.** *Let $\mathcal{P} = (\mathcal{T}, \mathcal{F})$ with query $q$ and $\mathcal{P}^{\sim} = (\mathcal{T}, \mathcal{F}, \mathcal{W})$ with $\mathcal{W} = \{:\sim q\ [C]\}$ with $C > 0$ or with $\mathcal{W} = \{:\sim not\ q\ [C]\}$ with $C < 0$. Then, $\underline{\mathrm{P}}^*(q) = \overline{\mathrm{P}}^*(q) = \underline{\mathrm{P}}(q)$.*

The credal semantics is not defined when there is at least one incoherent world. Incoherencies are usually due to constraints (or odd cycles through negation). The possibility of using *weak* constraints allows us to relax "strong" constraints, making such problems coherent and interpretable under the credal semantics.

**Example 5** (Maximal Clique in Probabilistic Graph)**.** *Let us now extend the program of Example 4 with the weak constraint of Example 3. This represents the maximal clique in a probabilistic graph problem, a well-known task studied in random graphs (Bläsius, Katzmann, and Stegehuis 2024).*

*We can now ask, for example, for the probability that a certain node is included in a* maximal *clique or for the probability that the* maximal *clique has a size greater than $k \in \mathbb{N}$. If the query is $in(1)$ (as in Example 4), $\underline{\mathrm{P}}^*(q) = 0.1856$ and $\overline{\mathrm{P}}^*(q) = 0.496$. For example, the world where all the edges are included has 12 possible answer sets but only 2 of them are optimal, and both have the query in it.*

**Example 6.** *Consider the program.*

```
0.2::a.
0.3::b.
q ; nq :- a.
q:- b.
:~ q. [-1]
```

*with query q. There are four worlds: $w_0 = \{\}$ has only one answer set, which is also optimal, without q, $w_1 = \{a\}$ has 2 answer sets, $\{a, q\}$ and $\{a, nq\}$, but only the first is optimal, so this contributes to $\underline{\mathrm{P}}^*(q)$ and $\overline{\mathrm{P}}^*(q)$, $w_2 = \{b\}$ and $w_3 = \{a, b\}$ have only one answer set with q, which is also optimal, so in both cases we have a contribution to $\underline{\mathrm{P}}^*(q)$ and $\overline{\mathrm{P}}^*(q)$. Overall, $\underline{\mathrm{P}}^*(q) = \overline{\mathrm{P}}^*(q) = 0.44$. If we ignore the weak constraint, $w_1$ contributes only to the upper probability, so $\overline{\mathrm{P}}(q) = 0.44$ while $\underline{\mathrm{P}}(q) = 0.3$. If the cost associated with the constraint is $> 0$, $w_1$ has only one optimal answer set without the query. So, in this case, the lower and upper probability would be 0.3.*

## 4 Modeling Examples

In this section, we demonstrate the modeling capabilities of optimal PASP by encoding three well-known problems within our framework.

**Optimal Reviewer Assignment.** The optimal reviewer assignment task is a well-studied problem (Bhaisare and Bharati 2025; Kalmukov 2020; Di Perro, Bernasconi, and Ferilli 2025; Amendola et al. 2016) and consists of assigning reviewers to articles for peer review. It has different flavors and variations. For ease of explanation, here we consider a simplified scenario in which each paper may be assigned to at least one reviewer. Each reviewer may or may not be available for a revision with a certain probability. The goal is to try to assign each paper to at least one reviewer such that the load among different reviewers is roughly the same. The following $\mathcal{P}^{\sim}$ models this problem.

```
paper(p1). paper(p2).
0.6::reviewer(r1). 0.7::reviewer(r2).
willing(R,P):- reviewer(R), paper(P).
assigned(R,P); skip(R,P) :- willing(R,P).
reviewed(P):- assigned(_,P).
load(R,2):-assigned(R,p1), assigned(R,p2).
load(R,1):-assigned(R,P), not load(R,2).
load(R,0):-reviewer(R), not load(R,1).
delta(R0,R1,D):- load(R0,L0), load(R1,L1),
    L0<=L1, D=L1-L0.
:~ paper(P), not reviewed(P). [1@1,P]
:~ delta(R0,R1,D). [D@0,R0,R1]
none_assigned :- load(R,0).
at_least_one_assigned :- not none_assigned.
```

The first two lines state that there are two papers to review, $p1$ and $p2$, and two possible reviewers, $r1$ and $r2$, which are available with a probability of 0.6 and 0.7, respectively. For simplicity, we assume that every reviewer is willing to review any paper ($willing/2$). A reviewer may ($assigned/2$) or may not ($skip/2$) be assigned to a paper if he/she is willing to review it. The load of a reviewer ($load/2$) is the number of papers he/she has to review. Lastly, $delta/3$ computes the difference in number of papers assigned between each pair of reviewers. There are two weak constraints: the first (at level 1) minimizes the number of papers not reviewed while the second (at level 0) minimizes the difference of load among reviewers. We may be interested in computing the probability that each reviewer is assigned to at lest one paper, by asking the probability of $at\_least\_one\_assigned$.

**Scheduling Problem under No-Show Behavior.** In the patient scheduling problem and overbooking under no-show behavior (Topuz et al. 2024; Cayirli, Yang, and Quek 2012; Alaeddini et al. 2011; Dodaro et al. 2021), patients should undergo different medical exams that require different resources. A patient may not show up, so the facility adopts overbooking to avoid leaving resources idle. However, this may result in patients that actually show up, but do not have the required resource assigned: in this case, they should be refunded. The following optimal PASP models the problem.

```
0.4::show(1). 0.4::show(2). 0.4::show(3).
0.4::show(4). 0.4::show(5).
resource(1,chair). resource(3,bed).
resource(2,chair). resource(4,bed).
exam(e1,chair). exam(e2,bed).
take(1,e1). take(2,e2). take(3,e1).
take(4,e2). take(5,e2).
require(P,R):- take(P,E),exam(E,R).
assign(P,R,Id) ; nAssign(P,R,Id) :- show(P),
     require(P,R), res(R,Id).
:- assign(P1,R,I), assign(P2,R,I), P1!=P2.
fulfilled(P):- show(P), assign(P,R,I).
refund_cost(P,2):-show(P), not fulfilled(P).
resource_cost(R,1):-assign(P1,chair,R).
resource_cost(R,2):-assign(P1,bed,R).
:~ refund_cost(R,C). [C@1,R]
:~ resource_cost(R,C). [C@0,R]
two_not_fulfilled:- show(A),show(B), A!=B,
   not fulfilled(A), not fulfilled(B).
```

Probabilistic facts $show(X)$ denote that the patient $X \in \{1, \dots, 5\}$ may or may not show up with probability 0.4 (for simplicity, we attach the same probability to each patient). Facts $exam(E, R)$ state that the exam $E$ requires the resource $R$. For simplicity, we assume that there are two exams, $e1$ and $e2$, and two types of resources, $chair$ and $bed$, each associated with an id ($resource(ID, R)$). Then, $take(P, E)$ states that the patient $P$ takes the exam $E$. A resource of $R$ with id $Id$ can be assigned ($assign(P, R, Id)$, and in this case, the request of the patient has been *fulfilled*) or not assigned ($nAssign(P, R, Id)$) to a patient $P$ who showed up and actually required that resource, and a resource cannot be assigned to two patients at the same time (constraint). The predicate $refound\_cost(P, R)$ states that the cost for refunding a patient $P$ that is present but not fulfilled is $R$. Similarly, used resources are associated with a cost through $resource\_cost/2$. Two weak constraints state that we want to minimize the cost of the refund (level 1) and then the cost of the used resources (level 0). In this scenario, we may be interested in computing the probability of having two patients not fulfilled, by querying $two\_not\_fulfilled$.

**Viral Marketing.** The viral marketing problem is a well-studied problem in decision theory (Van den Broeck et al. 2010; Domingos and Richardson 2001; Azzolini et al. 2025). Here, we have a network of people, represented by a graph. Some people may be targeted with an ad and could buy a particular product. Other people are not directly targeted by an ad, but they can buy the product as well, since they are influenced by someone who bought the product. The following optimal PASP, adapted from (Van den Broeck et al. 2010), models the problem:

```
market(P);noMarket(P) :- person(P).
0.7::buys_from_viral(P0,P1):-
   person(P0), person(P1), P0 != P1.
0.2::buys_from_marketing(P):-person(P).
buys(P):- market(P), buys_from_marketing(P).
buys(P):- friend(P,F), buys(F),
    buys_from_viral(P,F).
:~ market(P). [20@1,P]
:~ buys(P). [-5@1,P]
```

where, in addition, we have $person/1$ facts describing people in the network and $friend/2$ facts denoting friendship relations. We may decide to market ($market(P)$) or not ($nMarket(X)$) each person $P$. The rule with $0.7 ::$ $buys\_from\_viral(P0, P1)$ in the head is a syntactic sugar for writing a $0.7 ::$ $buys\_from\_viral(P0, P1)$ probabilistic fact for each pair of people $P1$ and $P2$. This states that the shopping behavior of $P1$ is influenced with probability 0.7 by the behavior of $P2$. Similarly for the probability of buying from direct marketing ($0.2 :: buys\_from\_marketing/1$). Then, a person $P$ buys ($buys(P)$) if he/she is marketed and buys from marketing or if he/she has a friend who bought the product and influences him/her. The two weak constraints, at the same priority level, state that marketing a particular person has a certain cost, but if someone buys the product, this gives a reward (i.e., the weak constraint is associated with a negative cost). Here, we can ask, for example, for the probability that a particular person buys a product. Moreover, by adding more involved constructs (Azzolini, Bellodi, and Riguzzi 2022), we may be able to ask queries such as whether at least 60% of the people will buy the product.

## 5 Complexity Results

We consider propositional programs and the following decision problems.

**Definition 4.** *Given an optimal PASP $\mathcal{P}^{\sim}$, a query $q$, and a rational $\gamma$, the*

- *brave inference task consists in checking whether $\overline{\mathrm{P}}^*(q) \geq \gamma$;*
- *cautious inference task consists in checking whether $\underline{\mathrm{P}}^*(q) \geq \gamma$.*

**Definition 5.** *Given an optimal PASP $\mathcal{P}^{\sim}$, an evidence $e$, and a rational $\gamma$, the*

- *brave MPE task consists in checking whether $P(\overline{\mathrm{MPE}}^*(e)) \geq \gamma$;*
- *cautious MPE task consists in checking whether $P(\underline{\mathrm{MPE}}^*(e)) \geq \gamma$.*

| | MPE / MPE* Complex. | | Inference Complex. |
|---|---|---|---|
| | $\overline{\mathrm{MPE}}(e)$ | $\underline{\mathrm{MPE}}(e)$ | $\overline{\mathrm{P}}(q)$ / $\overline{\mathrm{P}}^*(q)$ |
| Plain | $NP$ (Th. 4) | $\Sigma_2^P$ | $PP^{NP}$ |
| Weak Constr. | $\Sigma_2^P$ (Th. 9) | $\Sigma_2^P$ (Th. 9) | $C \cdot coNP$ (Th. 14) |

Table 1: Complexity results for plain and optimal PASP.

Complexity results for PASP without weak constraints (i.e., plain PASP) are well known. In particular, the authors of (Mauá and Cozman 2020) showed that the complexity of the cautious MPE problem for HCF normal programs (the fragment we consider in this paper) is $\Sigma_2^P$, while the complexity of cautious inference is $PP^{NP}$. Note that brave inference can be cast as cautious inference using the relation $\underline{\mathrm{P}}^*(q) = 1 - \overline{\mathrm{P}}^*(not\ q)$, so we can decide whether $\underline{\mathrm{P}}^*(not\ q) \leq 1 - \gamma$. However, this is not true for $\overline{\mathrm{MPE}}^*$ and $\underline{\mathrm{MPE}}^*$, therefore the two tasks must be studied separately.

We first show that brave and cautious cases have different complexity for plain PASP. Then, we focus on the MPE problem and show that, surprisingly, the complexity of the brave and cautious cases for optimal PASP coincides. Finally, we shift the attention to the brave inference task for optimal PASP (since the cautious case can be reconducted to the brave one). Table 1 summarizes the completeness results for both plain and optimal PASP.

### 5.1 MPE

Mauá and Cozman (2020) proved that the complexity of cautious MPE for plain PASP is complete for $\Sigma_2^P$. However, to the best of our knowledge, no completeness results are available for the brave MPE problem. To this end, the following theorem provides a completeness results for such a problem.

**Theorem 4.** *Let $\mathcal{P}$ be a plain PASP, $e$ be an evidence, and $\gamma \in [0, 1]$ be a rational number. Deciding whether $P(\overline{\mathrm{MPE}}(e)) \geq \gamma$ is NP-complete.*

*Proof (sketch).* (Hardness) Let $\Phi$ be a boolean formula in conjunctive normal form over a set of propositional variables $V$, then verifying the satisfiability of $\Phi$ is $NP$-complete. Such a problem can be encoded into a plain PASP $\mathcal{P}$ such that $P(\overline{\mathrm{MPE}}(e)) \geq 0.5^{|V|}$ iff $\Phi$ is satisfiable. More precisely, $\mathcal{P}$ is of the form $(\mathcal{T}, \mathcal{F})$, where $\mathcal{F}$ contains a probabilistic fact of the form $0.5 :: v$ for each $v \in V$ and $\mathcal{T}$ checks the satisfiability of $\Phi$. Intuitively, each selection $\sigma \subseteq \mathcal{F}$ encodes a possible truth assignment for variables in $V$ and its probability is $0.5^{|\mathcal{F}|} = 0.5^{|V|}$. Finally, $w_\sigma$ admits an answer set with evidence $e$ iff the truth assignment encoded by $\sigma$ satisfies $\Phi$. Thus, the thesis follows.

(Membership) Given a PASP $\mathcal{P} = (\mathcal{T}, \mathcal{F})$, we can encode the brave MPE problem with an HCF program $\mathcal{T}'$ that is coherent iff $P(\overline{\mathrm{MPE}}(e)) \geq \gamma$. Verifying the coherence of a heady-cycle-free ASP is $NP$-complete (Dantsin et al. 2001). More precisely, $\mathcal{T}'$ is made of rules of the form "$a\ ;\ na$" for each probabilistic fact $a$ (with $na$ being a fresh atom not appearing elsewhere) which guess a possible selection $\sigma \subseteq \mathcal{F}$. Then, rules and weak constraints in $\mathcal{T} \subseteq \mathcal{T}'$ guess an optimal answer set of the world $w_\sigma$, and finally two constraints ensure that the evidence $e$ is true in the guessed optimal answer set, and the probability of $\sigma$ is greater or equal than $\gamma$. Thus, $\mathcal{T}'$ admits an answer set iff there exists $\sigma \subseteq \mathcal{F}$ such that $P(\sigma) \geq \gamma$ and there exists $M \in AS^*(w_\sigma)$ such that $e \in M$, so $P(\overline{\mathrm{MPE}}(e)) \geq \gamma$. $\square$

Theorem 4 completes the complexity study on MPE for plain PASP, establishing that brave MPE is *easier* than the cautious MPE for plain PASP. We now study the complexity of MPE for optimal PASP.

First, we provide hardness for brave MPE via a reduction from the well-known Propositional Abduction Problem (PAP) (Eiter and Gottlob 1995a). Recall that a PAP is defined as a tuple of the form $\mathcal{A} = \langle V, T, H, M \rangle$, where $V$ is a set of variables, $T$ is a satisfiable propositional logic theory over variables in $V$, $H \subseteq V$ is a set of hypotheses (i.e., subset of the propositional variables), and $M \subseteq V$ is a set of manifestations (i.e., subset of the propositional variables). A *solution* for $\mathcal{A}$ is a set of the hypothesis $S \subseteq H$ such that $T \cup S$ is satisfiable and $T \cup S \vDash M$. Let $sol(\mathcal{A})$ be the set of solutions to PAP, then deciding whether $sol(\mathcal{A}) \neq \emptyset$ is $\Sigma_2^P$-complete (Eiter and Gottlob 1995a).

**Theorem 5** (Hardness brave MPE)**.** *Let $\mathcal{P}^\sim$ be an optimal PASP, $e$ be an evidence, and $\gamma \in [0, 1]$ be a rational number. Deciding whether $P(\overline{\mathrm{MPE}}^*(e)) \geq \gamma$ is $\Sigma_2^P$-hard.*

*Proof (sketch).* Let $\mathcal{A} = \langle V, T, H, M \rangle$ be a PAP, then we can encode $\mathcal{A}$ into an optimal PASP $\mathcal{P}^\sim = (\mathcal{T}, \mathcal{F}, \mathcal{W})$, where $\mathcal{F}$ contains a probabilistic fact of the form $0.5 :: s(x)$ for each hypothesis $x \in H$ and $\mathcal{T}$ checks whether a candidate solution $S$ is a solution for $\mathcal{A}$.

Intuitively, each selection $\sigma \subseteq \mathcal{F}$ represents a candidate solution $S = \{x \in H \mid s(x) \in \sigma\}$ for $\mathcal{A}$ and there exists $M \in AS^*(w_\sigma)$ such that $e \in M$ iff $S$ is a solution for $\mathcal{A}$.

Note that checking that $S$ is a solution for $\mathcal{A}$ requires verifying that $T \cup S$ is satisfiable and $T \cup S \vDash M$ holds. To verify that $T \cup S \vDash M$ holds, we need to ensure that for every truth assignment satisfying $T \cup S$, the manifestations in $M$ are satisfied as well. This is achieved by means of a weak constraint in $\mathcal{W}$ which adds a penalty if a truth assignment satisfies the entailment. In this way, if there exists an optimal answer set which satisfies the entailment then there exists no assignment that does not satisfy the entailment, and so $T \cup S \vDash M$ holds.

Finally, since each $\sigma \subseteq \mathcal{F}$ has the same probability $(0.5^{|H|})$, then $sol(\mathcal{A}) \neq \emptyset$ iff $P(\overline{\mathrm{MPE}}^*(e)) \geq 0.5^{|H|}$. $\square$

**Theorem 6** (Hardness cautious MPE)**.** *Let $\mathcal{P}^\sim$ be an optimal PASP, $e$ be an evidence, and $\gamma \in [0, 1]$ be a rational number. Deciding whether $P(\underline{\mathrm{MPE}}^*(e)) \geq \gamma$ is $\Sigma_2^P$-hard.*

*Proof.* The proof follows directly by observing that each PASP is indeed an optimal PASP where the underlying ASP contains no weak constraints. Deciding whether $P(\underline{\mathrm{MPE}}^*(e)) \geq \gamma$ for a PASP is $\Sigma_2^P$-hard (Mauá and Cozman 2020) and so the thesis follows. $\square$

From Theorem 5 and 6 we can observe that weak constraints cause a jump in complexity for brave MPE, but they do not introduce further complexity for cautious MPE. Indeed, both problems are hard for $\Sigma_2^P$.

We now provide memberships, which give us the completeness for cautious/brave MPE for optimal PASP. In both cases, we rely on ASP(Q) which allows modeling this problem in a very natural fashion.

**Theorem 7** (Membership brave MPE)**.** *Let $\mathcal{P}^\sim$ be an optimal PASP, $e$ be an evidence, and $\gamma \in [0, 1]$ be a rational number. Deciding whether $P(\overline{\mathrm{MPE}}^*(e)) \geq \gamma$ is in $\Sigma_2^P$.*

*Proof (sketch).* Given an optimal PASP $\mathcal{P}^\sim = (\mathcal{T}, \mathcal{F}, \mathcal{W})$ and an evidence $e$, verifying that $P(\overline{\mathrm{MPE}}^*(e)) \geq \gamma$ requires checking the existence of a selection $\sigma \subseteq \mathcal{F}$ such that $P(\sigma) \geq \gamma$ and there exists $M \in AS^*(w_\sigma)$ such that $e \in M$.

This can be encoded in ASP(Q) by a program $\Pi$ of the form $\exists^{st} \mathcal{T}_1 \exists^{st} \mathcal{T}_2 : C$, where the program $\mathcal{T}_1$ checks the existence of a selection $\sigma \subseteq \mathcal{F}$, $\mathcal{T}_2 = \mathcal{T} \cup \mathcal{W}$ checks the existence of an optimal answer set of $w_\sigma$, and finally the program $C$ imposes that evidence $e$ must be satisfied and $P(\sigma) \geq \gamma$. Thus, $\Pi$ is coherent iff $P(\overline{\mathrm{MPE}}^*(e)) \geq \gamma$. Verifying the coherence of $\Pi$ is $\Sigma_2^P$-complete (Mazzotta, Ricca, and Truszczynski 2024), and so the thesis follows. $\square$

Based on the approach of the proof (sketch) of Theorem 7 we can also provide a membership result for cautious MPE.

**Theorem 8** (Membership cautious MPE)**.** *Let $\mathcal{P}^\sim$ be an optimal PASP, $e$ be an evidence, and $\gamma \in [0, 1]$ be a rational number. Deciding whether $P(\underline{\mathrm{MPE}}^*(e)) \geq \gamma$ is in $\Sigma_2^P$.*

*Proof (sketch).* Given an optimal PASP $\mathcal{P}^\sim = (\mathcal{T}, \mathcal{F}, \mathcal{W})$, verifying that $P(\underline{\mathrm{MPE}}^*(e)) \geq \gamma$ requires checking the existence of a selection $\sigma \subseteq \mathcal{F}$ such that $P(\sigma) \geq \gamma$ and for each $M \in AS^*(w_\sigma)$, $e \in M$.

Also here we can build an ASP(Q) program $\Pi$ of the form $\exists^{st} \mathcal{T}_1 \exists^{st} \mathcal{T}_2 : C$, where the program $\mathcal{T}_1$ checks the existence of a selection $\sigma \subseteq \mathcal{F}$, then $\mathcal{T}_2$ checks the existence of an optimal answer set of $w_\sigma$, and finally the program $C$ imposes that evidence $e$ must be satisfied and $P(\sigma) \geq \gamma$.

To guarantee that $e$ is cautiously entailed (i.e., for every optimal answer set), we use a weak constraint in the program $\mathcal{T}_2$ which adds a penalty, at the lowest level, to each $M \in AS^*(w_\sigma)$ such that $e \in M$. In this way, if an optimal answer set of $\mathcal{T}_2$ satisfies evidence $e$, then $e$ is satisfied in every optimal answer set of $w_\sigma$. Thus, $\Pi$ is coherent iff $P(\underline{\mathrm{MPE}}^*(e)) \geq \gamma$. Verifying the coherence of $\Pi$ is $\Sigma_2^P$-complete (Mazzotta, Ricca, and Truszczynski 2024), and so the thesis follows. $\square$

**Theorem 9** (Complexity cautious and brave MPE)**.** *Given an optimal PASP $\mathcal{P}^\sim$, an evidence $e$, and a rational number*

$\gamma \in [0,1]$, *deciding whether* $(i)$ $P(\underline{\text{MPE}}^*(e)) \geq \gamma$ *is* $\Sigma_2^P$-*complete and* $(ii)$ $P(\overline{\text{MPE}}^*(e)) \geq \gamma$ *is* $\Sigma_2^P$-*complete.*

*Proof.* The proof follows from theorems 5-8. □

## 5.2 Inference

We now study the complexity of the brave inference problem. This problem can be reduced to a counting problem and so we rely on the counting complexity classes by Hemaspaandra and Vollmer (1995).

First, we recall the complexity of the brave reasoning for the ASP fragment considered in this paper (i.e., HCF programs with weak constraints), which will be instrumental in what follows.

**Theorem 10** (Buccafurri, Leone, and Rullo 2000)**.** *Let* $\mathcal{T}$ *be an HCF program with weak constraints and* $q$ *be an atom, then verify whether there exists* $M \in AS^*(\mathcal{T})$ *such that* $q \in M$ *is* $\Delta_2^P$-*complete.*

To prove hardness of inference in optimal PASP, we provide a reduction from the answer set counting problem for arbitrary ASP programs. To this end, we first recall the reduct-based stability check by Alviano et al. (2019) that will be instrumental to our reduction. Let $\mathcal{T}$ be an ASP and $I \subseteq B_{\mathcal{T}}$ be an interpretation, then $cl(\mathcal{T}, I) = \bigwedge_{r \in \mathcal{T}^I} cl(r, I)$, where $cl(r, I)$ denotes the disjunction:

$$(\bigvee_{a \in H(r) \cap I} a) \vee (\bigvee_{a \in B(r)^+} \neg a)$$

Moreover, $cl_{\subset}(I) = \bigvee_{a \in I} \neg a$ denotes the disjunction imposing that at least one of the true atoms w.r.t. $I$ should be false.

**Theorem 11** (Alviano et al. 2019)**.** *Let* $\mathcal{T}$ *be an ASP program, and* $I \subseteq B_{\mathcal{T}}$ *be an interpretation. Then* $I \in AS(\mathcal{T})$ *iff the formula* $cl(\mathcal{T}, I) \wedge cl_{\subset}(I)$ *is unsatisfiable.*

By exploiting this reduct-based stability check, it is possible to encode the answer counting problem for arbitrary ASP programs into an optimal PASP. To this end, we need to generalize the reduct-based stability check by Alviano et al. (2019) in such a way that $(i)$ the obtained formula does not depend on the current interpretation and $(ii)$ it is equivalent to $cl(P, I) \wedge cl_{\subset}(I)$ after that an interpretation $I$ is enforced.

**Definition 6.** *Let* $\mathcal{T}$ *be an ASP, then* $reduct(\mathcal{T}) = \bigwedge_{r \in \mathcal{T}} d_r$, *with* $d_r$ *being of the form:*

$$(\bigvee_{a \in H_r} a) \vee (\bigvee_{a \in B(r)^+} \neg a) \vee (\bigvee_{not\ a \in B(r)^-} a^p)$$

*where for each* $a \in B_{\mathcal{T}}$, $a^p$ *is a fresh atom.*

**Definition 7.** *Let* $\mathcal{T}$ *be an ASP, then* $subset(\mathcal{T})$ *denotes the following formula:*

$$(\bigwedge_{a \in B_{\mathcal{T}}} a^d \leftrightarrow a^p \wedge \neg a) \wedge (\bigvee_{a \in B_{\mathcal{T}}} a^d) \wedge (\bigwedge_{a \in B_{\mathcal{T}}} \neg a^p \rightarrow \neg a)$$

*where for each* $a \in B_{\mathcal{T}}$, $a^p$ *and* $a^d$ *are fresh atoms.*

**Lemma 1.** *Let* $\mathcal{T}$ *be an ASP, and* $I \subseteq B_{\mathcal{T}}$ *then* $cl(\mathcal{T}, I) \wedge cl_{\subseteq}(I)$ *is satisfiable iff* $reduct(\mathcal{T}) \wedge subset(\mathcal{T}) \wedge fix(I)$ *is satisfiable, where* $fix(I)$ *is the following formula:*

$$(\bigwedge_{a \in I} a^p) \wedge \bigwedge_{a \in B_{\mathcal{T}} \setminus I} \neg a^p$$

*Proof (sketch).* Let $I \subseteq B_{\mathcal{T}}$ and $\phi_I = reduct(\mathcal{T}) \wedge subset(\mathcal{T}) \wedge fix(I)$. Intuitively, the subformula $fix(I)$ enforces the interpretation $I$ in $\phi_I$ by means of fresh propositional variables of the form $a^p$ for each $a \in B_{\mathcal{T}}$. Thus, by construction, each truth assignment satisfying $\phi_I$, is such that $a^p$ is true iff $a$ is true w.r.t. $I$. This means that the formula $\phi_I$ can be simplified accordingly. After this simplification, it is possible to observe that the obtained formula is equal to $cl(\mathcal{T}, I) \wedge cl_{\subseteq}(I)$ modulo variable renaming, and so the thesis follows. □

From Lemma 1, it is possible to encode the answer set counting problem into an optimal PASP. The answer set counting problem is $\# \cdot coNP$-complete (Fichte et al. 2017). Thus, from Hemaspaandra and Vollmer (1995), the decisional counterpart of such a problem (i.e., deciding whether an ASP $\mathcal{T}$ admits at least $k$ answer sets) is $C \cdot coNP$-complete. Based on these observations, we reduce the answer set counting problem to the brave inference task for optimal PASP, which gives the hardness result.

**Theorem 12** (Hardness brave inference)**.** *Let* $\mathcal{P}^{\sim}$ *be an optimal PASP,* $q$ *be an atom, and* $\gamma$ *be a rational number in* $[0,1]$, *then verifying whether* $\overline{\text{P}}^*(q) \geq \gamma$ *is hard for* $C \cdot coNP$.

*Proof (sketch).* Given an ASP $\mathcal{T}$ and an integer $k$, it is possible to construct an optimal PASP $\mathcal{P}^{\sim} = (\mathcal{T}, \mathcal{F}, \mathcal{W})$ such that $\overline{\text{P}}^*(unsat) \geq k \cdot 0.5^{|B_{\mathcal{T}}|}$ iff $|AS(\mathcal{T})| \geq k$.

More precisely, $\mathcal{F}$ contains a probabilistic fact of the form $0.5 :: a$ for each $a \in B_{\mathcal{T}}$ and so, each selection $\sigma \subseteq \mathcal{F}$ is indeed an interpretation $I$ over $B_{\mathcal{T}}$ which has an associated probability of $0.5^{|B_{\mathcal{T}}|}$. Then, $\mathcal{T}$ guesses a truth assignment for the variables in $\phi_I$ and checks whether it satisfies $\phi_I$ or not. Finally, $\mathcal{W}$ contains a weak constraint which adds a penalty to the assignments that do not satisfy $\phi_I$. Thus, the truth assignments that satisfy $\phi_I$ are preferred. This means that an optimal answer set contains the query $unsat$ (i.e. the guessed truth assignment does not satisfy $\phi_I$) iff $\phi_I$ is unsatisfiable and so, $\overline{\text{P}}^*(unsat) \geq k \cdot 0.5^{|B_{\mathcal{T}}|}$ iff there exists at least $k$ answer set. Thus, the thesis follows. □

We now provide membership results.

**Theorem 13** (Membership brave inference)**.** *Let* $\mathcal{P}^{\sim}$ *be an optimal PASP,* $q$ *be an atom, and* $\gamma \in [0,1]$ *be a rational number. Then, verifying whether* $\overline{\text{P}}^*(q) \geq \gamma$ *is in* $C \cdot coNP$.

*Proof.* With slight modifications, we can apply the methodology employed by Cozman and Mauá (2017) for proving the complexity of brave inference for plain PASP.

Let $\mathcal{P}^{\sim} = (\mathcal{T}, \mathcal{F}, \mathcal{W})$ be an optimal PASP. We can construct a non-deterministic Turing Machine $TM_{\mathcal{P}^{\sim}}$ such that for each $p_i :: a_i \in \mathcal{F}$, with $p_i = \mu_i / \nu_i$ for some (smallest) integers $\mu_i$ and $\nu_i$, it chooses one among $\nu_i$ computation paths. For $\mu_i$ of these computation paths, $a_i$ is considered true in the current selection while, for the $\nu_i - \mu_i$ remaining ones, $a_i$ is considered false. This means that such a machine has $\prod_i \nu_i$ computation paths, and for each selection $\sigma \subseteq \mathcal{F}$ there are $\prod_{a_i \in \sigma} \mu_i \cdot \prod_{a_i \in \mathcal{F} \setminus \sigma} (\nu_i - \mu_i)$ computation paths. Each of these paths is polynomial w.r.t. the input size and returns yes iff $\exists M \in AS^*(w_\sigma)$ such that $q \in M$. From Theorem 10, verifying the existence of such $M$ is $\Delta_2^P$-complete problem. Thus, from Definition 1.5 of Hemaspaandra and Vollmer (1995), the function $f$ that count the number of accepting path of $TM_{\mathcal{P}^\sim}$ belongs to $\# \cdot \Delta_2^P = \# \cdot P^{NP}$. Moreover, from Theorem 1.5 of Hemaspaandra and Vollmer (1995) $\# \cdot P^{NP} = \# \cdot coNP$. Thus, $f \in \# \cdot coNP$. Let $\mathcal{S} = \{\sigma \subseteq \mathcal{F} \mid \exists M \in AS^*(w_\sigma), q \in M\}$ be the set of selection which bravely satisfies the query $q$, then the number of accepting paths of $TM_{\mathcal{P}^\sim}$ is:

$$f(\mathcal{P}^\sim) = \sum_{\sigma \in \mathcal{S}} \left( \prod_{a_i \in \sigma} \mu_i \cdot \prod_{a_i \in \mathcal{F} \setminus \sigma} (\nu_i - \mu_i) \right)$$

Let $g(\mathcal{P}^\sim) = \gamma \cdot ( \prod_{p_i :: a_i \in \mathcal{P}^\sim,\ p_i = \mu_i / \nu_i} \nu_i)$ then, $\overline{\mathrm{P}}^*(q) \geq \gamma$ iff $f(\mathcal{P}^\sim) \geq g(\mathcal{P}^\sim)$, and so, the problem is in $C \cdot coNP$. □

**Theorem 14.** *Let $\mathcal{P}^\sim$ be an optimal PASP, $q$ be an atom, and $\gamma \in [0, 1]$ be a rational number. Then, verifying whether $\overline{\mathrm{P}}^*(q) \geq \gamma$ is $C \cdot coNP$-complete.*

*Proof.* The thesis follows from Theorem 12 and 13. □

## 6 Related Work

For stratified programs, where every world has a unique stable model (which also coincides with the well-founded model), there exist many approaches (Sato 1995; Muggleton 2003; Meert, Struyf, and Blockeel 2010; Van den Broeck et al. 2010; Riguzzi and Swift 2011), although none of these considers weak constraints or any other optimality criteria. The authors of (Azzolini and Riguzzi 2021) proposed a framework based on the distribution semantics (so, one model per world) where each probabilistic fact has an associated range and the goal is to find the optimal value within this range such that the probability of a query is maximized (which is similar to the task of parameter learning (Riguzzi 2022)). This is different from our approach, as we consider optimality over answer sets and do not aim at finding a probability value within a given range.

If we consider probabilistic extensions of ASP, the credal semantics and the L-credal semantics (Rocha and Gagliardi Cozman 2022), which extends the CS by lifting the requirement of having at least one model per world (but introduces a third "undefined" truth value), are the most similar to our proposal. However, for both, no optimality statements are considered. In this paper, we extend the CS with weak constraints. An interesting future work could be to frame our approach in the context of the L-credal semantics. The smProbLog semantics (Totis, De Raedt, and Kimmig 2023) focuses on normal ASP (so no weak constraints) extended with probabilistic facts. However, instead of considering the contribution of a world to the lower and/or upper bounds, it assumes a uniform probability distribution over the answer sets of a world (in line with the maximum entropy principle). That is, each world $w$ contributes to the probability of a query with a value of $P(w)$ (computed as in Equation 1) divided by the number of answer sets of $w$ in which the query is present. This is a specialization of the CS, since the CS assumes no probability distribution over the answer sets. Thus, the probability of a query computed under the smProbLog semantics always falls within the lower and upper bounds defined by the CS. Furthermore, the smProbLog semantics considers a third truth value, "inconsistent", to also support worlds without models.

Another semantics, though substantially different from our approach, is LPMLN (Lee, Talsania, and Wang 2017). An LPMLN program consists of normal ASP rules associated with a weight (so $\in \mathbb{R}$ instead of $\in [0, 1]$). Those rules are called "soft", in contrast to hard rules, which are associated with infinite weight. Each stable model is assigned a weight and then a probability by dividing its weight by the sum of all weights. The probability of the query is the sum of the probabilities of the models containing the query. Differently from us (and from the CS), LPMLN does not have the distinction between world and model for each world; rather, it considers only the stable models of a program. This makes it easier to compute, for example, the most probable answer set (which the authors call MAP, but usually MAP in statistical relational artificial intelligence is a different and substantially harder problem), since weak constraints can be used off-the-shelf (Lee, Talsania, and Wang 2017). Note that LPMLN considers weak constraints only for the computation of such an answer set and not as part of the program.

## 7 Conclusions

In this paper, we extend the credal semantics for Probabilistic Answer Set Programming to handle weak constraints, thus introducing the *optimal* PASP framework. We motivate our proposal by encoding, in an easy and intuitive way, three real-world problems requiring to reason on both optimality criteria and probabilities, namely: Optimal Reviewer Assignment, Scheduling Problem under No Show Behaviour, and Viral Marketing. Moreover, we formalized the inference and MPE tasks and provided a complexity study, precisely placing them in the complexity landscape.

As a future work, we plan to develop a complexity-aware solver for handling such programs, e.g., resorting to knowledge compilation (Darwiche and Marquis 2002) and/or ASP(Q) (Mazzotta, Ricca, and Truszczynski 2024).

## Acknowledgments

This work has been partially supported by Spoke 1 "FutureHPC & BigData" of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 - Next Generation EU (NGEU), by the Italian Ministry of Industrial Development (MISE) under project EI-TWIN n. F/310168/05/X56 CUP B29J24000680005, and by the Italian Ministry of Research (MUR) under projects: PNRR FAIR - Spoke 9 - WP 9.1 CUP H23C22000860006, Tech4You CUP H23C22000370006, and PRIN PINPOINT

CUP H23C22000280006. DA and FR are members of the Gruppo Nazionale Calcolo Scientifico – Istituto Nazionale di Alta Matematica (GNCS-INdAM).

## References

Alaeddini, A.; Yang, K.; Reddy, C.; and Yu, S. 2011. A probabilistic model for predicting the probability of no-show in hospital appointments. *Health care management science* 14:146–157.

Alviano, M.; Calimeri, F.; Dodaro, C.; Fuscà, D.; Leone, N.; Perri, S.; Ricca, F.; Veltri, P.; and Zangari, J. 2017. The ASP system DLV2. In *LPNMR*, volume 10377 of *Lecture Notes in Computer Science*, 215–221. Springer.

Alviano, M.; Amendola, G.; Dodaro, C.; Leone, N.; Maratea, M.; and Ricca, F. 2019. Evaluation of disjunctive programs in WASP. In *LPNMR*, volume 11481 of *Lecture Notes in Computer Science*, 241–255. Springer.

Amendola, G.; Dodaro, C.; Leone, N.; and Ricca, F. 2016. On the application of answer set programming to the conference paper assignment problem. In *AI*IA*, volume 10037 of *Lecture Notes in Computer Science*, 164–178. Springer.

Amendola, G.; Ricca, F.; and Truszczynski, M. 2019. Beyond NP: quantifying over answer sets. *Theory and Practice of Logic Programming* 19(5-6):705–721.

Azzolini, D., and Riguzzi, F. 2021. Optimizing probabilities in probabilistic logic programs. *Theory and Practice of Logic Programming* 21(5):543–556.

Azzolini, D.; Bellodi, E.; and Riguzzi, F. 2022. Statistical statements in probabilistic logic programming. In Gottlob, G.; Inclezan, D.; and Maratea, M., eds., *Logic Programming and Nonmonotonic Reasoning*, 43–55. Cham: Springer International Publishing.

Azzolini, D.; Bellodi, E.; Kiesel, R.; and Riguzzi, F. 2025. Solving decision theory problems with probabilistic answer set programming. *Theory and Practice of Logic Programming* 25(1):33–63.

Bhaisare, B., and Bharati, R. 2025. Advancing peer review integrity: Automated reviewer assignment techniques with a focus on deep learning applications. In Bairwa, A. K.; Tiwari, V.; Vishwakarma, S. K.; Tuba, M.; and Ganokratanaa, T., eds., *Computation of Artificial Intelligence and Machine Learning*, 312–327. Cham: Springer Nature Switzerland.

Bläsius, T.; Katzmann, M.; and Stegehuis, C. 2024. Maximal cliques in scale-free random graphs. *Network Science* 12(4):366–391.

Brewka, G.; Eiter, T.; and Truszczyński, M. 2011. Answer set programming at a glance. *Communications of the ACM* 54(12):92–103.

Buccafurri, F.; Leone, N.; and Rullo, P. 2000. Enhancing disjunctive datalog by constraints. *IEEE Transactions on Knowledge and Data Engineering* 12(5):845–860.

Calimeri, F.; Faber, W.; Gebser, M.; Ianni, G.; Kaminski, R.; Krennwallner, T.; Leone, N.; Maratea, M.; Ricca, F.; and Schaub, T. 2020. ASP-Core-2 input language format. *Theory and Practice of Logic Programming* 20(2):294–309.

Cayirli, T.; Yang, K.; and Quek, S. 2012. A universal appointment rule in the presence of no-shows and walk-ins. *Production and Operations Management* 21.

Cozman, F. G., and Mauá, D. D. 2017. On the semantics and complexity of probabilistic logic programs. *Journal of Artificial Intelligence Research.* 60:221–262.

Cozman, F. G., and Mauá, D. D. 2020. The joy of probabilistic answer set programming: Semantics, complexity, expressivity, inference. *International Journal of Approximate Reasoning* 125:218–239.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys* 33(3):374–425.

Darwiche, A., and Marquis, P. 2002. A knowledge compilation map. *Journal of Artificial Intelligence Research* 17:229–264.

De Raedt, L.; Kimmig, A.; and Toivonen, H. 2007. ProbLog: A probabilistic Prolog and its application in link discovery. In Veloso, M. M., ed., *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, volume 7, 2462–2467. AAAI Press.

Di Perro, D.; Bernasconi, E.; and Ferilli, S. 2025. Automatic reviews' assignments through ansewer set programming. In *IRCDL 2025: 21st Conference on Information and Research Science Connecting to Digital and Library Science, February 20-21 2025, Udine, Italy*.

Dix, J.; Gottlob, G.; and Marek, V. W. 1996. Reducing disjunctive to non-disjunctive semantics by shift-operations. *Fundam. Informaticae* 28(1-2):87–100.

Dodaro, C.; Galata, G.; Grioni, A.; Maratea, M.; Mochi, M.; and Porro, I. 2021. An ASP-based solution to the chemotherapy treatment scheduling problem. *Theory and Practice of Logic Programming* 21(6):835–851.

Domingos, P., and Richardson, M. 2001. Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 57–66. New York, NY, USA: Association for Computing Machinery.

Durand, A.; Hermann, M.; and Kolaitis, P. G. 2000. Subtractive reductions and complete problems for counting complexity classes. In *MFCS*, volume 1893 of *Lecture Notes in Computer Science*, 323–332. Springer.

Eiter, T., and Gottlob, G. 1995a. The complexity of logic-based abduction. *Journal of the ACM* 42(1):3–42.

Eiter, T., and Gottlob, G. 1995b. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence* 15(3-4):289–323.

Erdem, E.; Gelfond, M.; and Leone, N. 2016. Applications of answer set programming. *AI Magazine* 37(3):53–68.

Faber, W.; Pfeifer, G.; and Leone, N. 2011. Semantics and complexity of recursive aggregates in answer set programming. *Artificial Intelligence* 175(1):278–298.

Falkner, A. A.; Friedrich, G.; Schekotihin, K.; Taupe, R.; and Teppan, E. C. 2018. Industrial applications of answer set programming. *Künstliche Intelligenz* 32(2-3):165–176.

Fichte, J. K.; Hecher, M.; Morak, M.; and Woltran, S. 2017. Answer set solving with bounded treewidth revisited. In Balduccini, M., and Janhunen, T., eds., *Logic Programming and Nonmonotonic Reasoning*, 132–145. Cham: Springer International Publishing.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2019. Multi-shot ASP solving with clingo. *Theory and Practice of Logic Programming* 19(1):27–82.

Hemaspaandra, L. A., and Vollmer, H. 1995. The satanic notations: counting classes beyond #P and other definitional adventures. *SIGACT News* 26(1):2–13.

Kalmukov, Y. 2020. An algorithm for automatic assignment of reviewers to papers. *Scientometrics* 124(3):1811–1850.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. Cambridge, MA: MIT Press.

Lee, J.; Talsania, S.; and Wang, Y. 2017. Computing LPMLN using ASP and MLN solvers. *Theory and Practice of Logic Programming* 17(5-6):942–960.

Mauá, D. D., and Cozman, F. G. 2020. Complexity results for probabilistic answer set programming. *International Journal of Approximate Reasoning* 118:133–154.

Mazzotta, G.; Ricca, F.; and Truszczynski, M. 2024. Quantifying over optimum answer sets. *Theory and Practice of Logic Programming* 24(4):716–736.

Meert, W.; Struyf, J.; and Blockeel, H. 2010. CP-Logic theory inference with contextual variable elimination and comparison to BDD based inference methods. In De Raedt, L., ed., *Inductive Logic Programming, 19th International Conference, (ILP 2009)*, volume 5989 of *Lecture Notes on Computer Science*, 96–109. Springer.

Muggleton, S. 2003. Learning structure and parameters of stochastic logic programs. In *Inductive Logic Programming: 12th International Conference, ILP 2002 Sydney, Australia, July 9–11, 2002 Revised Papers 12*, 198–206. Springer.

Papadimitriou, C. H. 1994. *Computational complexity*. Addison-Wesley.

Raedt, L. D.; Kersting, K.; Natarajan, S.; and Poole, D. 2016. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 10(2):1–189.

Riguzzi, F., and Swift, T. 2011. The PITA system: Tabling and answer subsumption for reasoning under uncertainty. *Theory and Practice of Logic Programming* 11(4-5):433–449.

Riguzzi, F. 2022. *Foundations of Probabilistic Logic Programming Languages, Semantics, Inference and Learning, Second Edition*. Gistrup, Denmark: River Publishers.

Rocha, V. H. N., and Gagliardi Cozman, F. 2022. A credal least undefined stable semantics for probabilistic logic programs and probabilistic argumentation. In Kern-Isberner, G.; Lakemeyer, G.; and Meyer, T., eds., *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022*, 309–319.

Sato, T. 1995. A statistical learning method for logic programs with distribution semantics. In Sterling, L., ed., *Logic Programming, Proceedings of the Twelfth International Conference on Logic Programming*, 715–729. MIT Press.

Stockmeyer, L. J. 1976. The polynomial-time hierarchy. *Theoretical Computer Science* 3(1):1–22.

Topuz, K.; Urban, T. L.; Russell, R. A.; and Yildirim, M. B. 2024. Decision support system for appointment scheduling and overbooking under patient no-show behavior. *Annals of Operations Research* 342(1):845–873.

Totis, P.; De Raedt, L.; and Kimmig, A. 2023. smProbLog: Stable model semantics in problog for probabilistic argumentation. *Theory and Practice of Logic Programming* 1–50.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *SIAM Journal on Computing* 8(3):410–421.

Van den Broeck, G.; Thon, I.; van Otterlo, M.; and De Raedt, L. 2010. DTProbLog: A decision-theoretic probabilistic Prolog. In Fox, M., and Poole, D., eds., *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1217–1222. AAAI Press.

Van Gelder, A.; Ross, K. A.; and Schlipf, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM* 38(3):620–650.