

Bridging Uncertainty and Optimization in Probabilistic ASP

Damiano Azzolini¹, Giuseppe Mazzotta², Francesco Ricca² and Fabrizio Riguzzi¹

¹University of Ferrara, Italy

²University of Calabria, Italy

Abstract

Probabilistic Answer Set Programming (PASP) combines the expressiveness of Answer Set Programming with probabilistic reasoning, providing a powerful framework for representing uncertain knowledge. However, current PASP semantics do not support optimization criteria, limiting their applicability in real-world decision-making scenarios where uncertainty and optimization naturally coexist. In the paper entitled “A Novel Framework for Reasoning over Optimization Problems in Probabilistic Answer Set Programming” and published in the proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning (KR2025), we introduced Optimal Probabilistic Answer Set Programming (OptPASP), an extension of PASP that integrates weak constraints into the probabilistic framework. In this discussion paper, we overview the proposed framework, highlight its modeling capabilities, discuss its computational complexity, and outline promising research directions.

Keywords

Answer Set Programming, Probabilistic ASP, Uncertainty, Optimization

1. Introduction

Reasoning and decision making under uncertainty are central challenges in Artificial Intelligence. Many real-world applications, including resource allocation [1], scheduling [2, 3], robotics [4], query answering [5], among others [6], requiring simultaneously handling uncertain information and optimizing one or more objectives. While probabilistic models provide principled mechanisms for representing uncertainty, declarative reasoning frameworks offer powerful tools for expressing complex constraints and optimization criteria. Combining these two perspectives remains an important research direction.

Probabilistic Logic Programming (PLP) has long provided effective frameworks for integrating logical reasoning and probability [7, 8]. At the same time, Answer Set Programming (ASP) has established itself as one of the most successful declarative paradigms for solving combinatorial search and optimization problems, thanks to its expressive modeling language and highly efficient solvers [9, 10]. Probabilistic Answer Set Programming (PASP) [11] represents a significant step toward combining these two worlds, extending ASP with probabilistic reasoning.

Despite its expressive power, current PASP frameworks offer limited support for optimization. In particular, they do not naturally accommodate weak constraints, which are the standard mechanism for expressing preferences and optimization criteria in ASP. This limitation reduces the applicability of PASP in many AI scenarios where uncertainty and optimization are tightly coupled. For example, assigning resources under uncertain availability, planning under uncertain demand, or selecting actions that maximize expected benefits all require reasoning about both aspects simultaneously.

To address this gap, we introduced *Optimal Probabilistic Answer Set Programming* (OptPASP) [12], an extension of PASP that allows the usage of weak constraints. The resulting framework enables reasoning over optimal solutions in uncertain environments while preserving the declarative nature of ASP. In this paper we overview the proposed framework and discuss interesting real-world problems that can

AIxIA 2026 Discussion Papers, 24th International Conference of the Italian Association for Artificial Intelligence, Perugia, Italy, 2026

✉ damiano.azzolini@unife.it (D. Azzolini); giuseppe.mazzotta@unical.it (G. Mazzotta); francesco.ricca@unical.it (F. Ricca); fabrizio.riguzzi@unife.it (F. Riguzzi)

0000-0002-7133-2673 (D. Azzolini); 0000-0003-0125-0477 (G. Mazzotta); 0000-0001-8218-3178 (F. Ricca); 0000-0003-1654-9703 (F. Riguzzi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

be modeled in OptPASP. Finally, we analyze the computational complexity of the main reasoning task in OptPASP which gives theoretical boundaries on the expressiveness of such formalism.

2. Optimal Probabilistic ASP

Probabilistic ASP. We briefly recall the notions of Answer Set Programming (ASP) and Probabilistic Answer Set Programming (PASP) that are relevant to this work. For comprehensive definitions and formal details, we refer the reader to the ASP-Core standard [13] and works on probabilistic logic programming and PASP [11, 14].

An ASP consists of a finite set of rules and weak constraints. The semantics of a program is given by its answer sets, i.e., minimal models of its FLP reduct [15]. Weak constraints induce a preference relation among answer sets through a minimization of violation costs. We denote by $AS(\mathcal{T})$ and $AS^*(\mathcal{T})$ the sets of answer sets and optimal answer sets of a program $\mathcal{T}$, respectively.

A probabilistic answer set program (PASP) [14] is a pair $\mathcal{P} = (\mathcal{T}, \mathcal{F})$, where $\mathcal{T}$ is a head-cycle-free ASP program without weak constraints (i.e., plain HCF ASP) and $\mathcal{F}$ is a set of probabilistic facts of the form $\pi :: a$, with $\pi \in [0, 1]$. Each subset $\sigma \subseteq \mathcal{F}$ (called selection) induces a world w_σ , obtained by adding to $\mathcal{T}$ the atoms of the probabilistic facts selected in σ . Independence among probabilistic facts is assumed and the probability of a world is

$$P(w_\sigma) = \prod_{\pi_i :: a_i \in \mathcal{F} | a_i \in w_\sigma} \pi_i \cdot \prod_{\pi_i :: a_i \in \mathcal{F} | a_i \notin w_\sigma} (1 - \pi_i).$$

Under the credal semantics [11], every world is required to have at least one answer set (i.e., to be coherent). Given a query atom q , its lower and upper probabilities are defined as

$$\underline{P}(q) = \sum_{w_\sigma | \forall M \in AS(w_\sigma), M \models q} P(w_\sigma), \quad \overline{P}(q) = \sum_{w_\sigma | \exists M \in AS(w_\sigma), M \models q} P(w_\sigma).$$

Another common task is the computation of the Most Probable Explanation (MPE), which identifies the most probable world in which an evidence atom is cautiously or bravely entailed. Formally,

$$\underline{\text{MPE}}(e) = \arg \max_{\sigma | \forall M \in AS(w_\sigma), M \models e} P(w_\sigma), \quad \overline{\text{MPE}}(e) = \arg \max_{\sigma | \exists M \in AS(w_\sigma), M \models e} P(w_\sigma).$$

Extending PASP with Weak Constraints PASP combines probabilistic reasoning with the expressive modeling capabilities of ASP by generating a set of possible worlds from probabilistic facts and performing reasoning over the answer sets associated with each world. Under this perspective, every world is an ASP. However, while optimization is a central feature of ASP and is routinely used to model preference and decision-making problems, its role in probabilistic ASP has received little attention.

This observation naturally motivates the extension considered in this paper. Since each world is an ASP, it is possible to equip worlds with weak constraints and reason over their optimal answer sets. This yields a probabilistic framework where uncertainty and optimization coexist: probabilistic facts determine which worlds are considered, while weak constraints determine which solutions are preferred within each world.

Definition 1. An optimal probabilistic answer set program (*optimal PASP*) is a tuple $\mathcal{P}^\sim = (\mathcal{T}, \mathcal{F}, \mathcal{W})$, 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.

Given a selection σ , the corresponding world w_σ is obtained as in PASP. Weak constraints do not affect the probability of a world, but only the answer sets considered in that world. Therefore, for each world w_σ , we consider the set of optimal answer sets $AS^*(w_\sigma)$ instead of $AS(w_\sigma)$. As in standard PASP under the credal semantics, we require every world to be coherent.

The lower and upper probabilities of a query q are defined as

$$\underline{P}(q) = \sum_{w_\sigma | \forall M \in AS^*(w_\sigma), M \models q} P(w_\sigma), \quad \overline{P}(q) = \sum_{w_\sigma | \exists M \in AS^*(w_\sigma), M \models q} P(w_\sigma).$$

Similarly, the lower and upper MPE states for an evidence atom e are

$$\underline{\text{MPE}}(e) = \arg \max_{\sigma | \forall M \in AS^*(w_\sigma), M \models e} P(w_\sigma), \quad \overline{\text{MPE}}(e) = \arg \max_{\sigma | \exists M \in AS^*(w_\sigma), M \models e} P(w_\sigma).$$

3. Modeling Optimization under Uncertainty

A major advantage of OptPASP is its modeling flexibility. The framework naturally captures problems where uncertainty and optimization interact. In [12], we modeled three interesting problems from different fields which are Viral Marketing [16, 17, 18], Optimal Reviewer Assignment [19, 20, 21, 22] and Scheduling under No-Show Behaviour [23, 24, 25, 26]. Here we only discuss the Scheduling under No-Show Behaviour which can easily be adapted into other domains such as event booking.

Scheduling under No-Show Behaviour. In patient scheduling and overbooking problems under no-show behavior [23, 24, 25, 26], patients require different medical examinations, each consuming specific resources. Since patients may fail to attend their appointments, healthcare facilities may adopt overbooking strategies to improve resource utilization. However, excessive overbooking may result in situations where patients who actually show up cannot be served because the required resources are unavailable. In such cases, the facility incurs a refund cost.

This setting naturally combines uncertainty and optimization. On the one hand, patient attendance is uncertain and can be modeled with probabilistic facts. On the other hand, the facility aims to allocate resources so as to minimize refund costs while limiting resource utilization. An important question is therefore whether the adopted overbooking policy is sufficiently robust. For example, one may be interested in computing the probability that some patients who show up will not be served under an optimal scheduling policy.

This reasoning can be easily modelled in the OptPASP framework, as follows.

```
0.4::show(1). 0.4::show(2). 0.4::show(3). 0.4::show(4). 0.4::show(5).
resource(1,chair). resource(2,chair). resource(3,bed). 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), resource(Id,R).
:- 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 of the form $show(P)$ model the uncertainty associated with patient attendance. In the example, each of the five patients shows up with probability 0.4. Facts $exam(E, R)$ specify the type of resource required by an examination, while $take(P, E)$ associates patients with the examinations they are scheduled to undergo.

Whenever a patient is present and requires a resource, the program decides whether a specific resource instance is assigned to that patient. A constraint ensures that the same resource cannot be assigned to multiple patients simultaneously. A patient is considered *fulfilled* if the required resource is successfully allocated; otherwise, a refund cost is incurred. In addition, using resources is associated with an operational cost.

The optimization criterion is expressed through two weak constraints. The primary objective (priority level 1) is to minimize refund costs, whereas the secondary objective (priority level 0) is to minimize the cost of allocated resources. Thus, among all feasible allocations, the framework first seeks to maximize patient satisfaction and then to minimize resource utilization costs.

A possible query in this setting is *two_not_fulfilled*, which represents the event that at least two patients who show up cannot be served.

4. Discussion

The introduction of optimization into Probabilistic Answer Set Programming naturally raises questions about computational complexity. In ASP, reasoning over optimal answer sets is typically harder than reasoning over arbitrary answer sets, since computing an optimal solution generally requires a polynomial number of calls to an NP oracle. Consequently, one may expect that extending PASP with weak constraints would lead to a corresponding increase in the complexity of its main reasoning tasks.

Our analysis [12] shows that this intuition is only partially correct. For probabilistic inference, the introduction of weak constraints does not increase the computational complexity, and the problem remains complete for PP^{NP} [27]. Intuitively, inference already requires aggregating information across an exponential number of possible worlds. The additional effort needed to identify optimal answer sets within each world is therefore absorbed by the counting process underlying probability computation.

A different picture emerges for the Most Probable Explanation (MPE). In particular, brave MPE becomes harder, moving from NP -complete for plain PASP to Σ_2^P -complete in the presence of optimization. By contrast, cautious MPE remains Σ_2^P -complete, yielding the same complexity as in the non-optimized setting. As a consequence, once weak constraints are introduced, brave and cautious MPE become computationally equivalent.

This difference can be explained by the nature of the underlying ASP reasoning tasks. Brave MPE is based on the existence of an answer set satisfying a given query. In plain PASP, this amounts to checking whether the query holds in some answer set of a world which is easier than performing the same check over optimal answer sets only.

Conversely, cautious MPE already relies on a universal quantification over answer sets. Replacing all answer sets with the subset of optimal ones does not introduce any computational overhead and so no jump in complexity is introduced.

Beyond complexity considerations, these results highlight several research directions. From a practical perspective, the development of efficient OptPASP solvers is a crucial step toward the adoption of the framework in real-world applications. For probabilistic inference, promising directions include the integration of knowledge compilation techniques [28], whereas, for MPE, it would be interesting to investigate whether recent extension of ASP, such as Answer Set Programming with Quantifiers [29, 30], can be exploited to obtain more efficient solving strategies [31, 32, 33].

From a theoretical perspective, several questions remain open. Alternative probabilistic semantics [34, 35, 36] could be considered, as well as richer optimization constructs [37] beyond weak constraints. Moreover, studying the relationship between OptPASP and decision-theoretic probabilistic logic frameworks [18] may provide further insights into the role of optimization under uncertainty.

Acknowledgments

This work has been partially funded by the Italian Ministry of Industrial Development (MISE) under project EI-TWIN n. F/310168/05/X56 CUP B29J24000680005 and also partially by projects SIGENERA (CUP J29I24001770005) and MOZART (J49I24001740005) selected within the framework of the PR FESR – FSE Calabria 2021/2027 and implemented with the support of the Italian State and the Calabria Region. DA and FR are members of the Gruppo Nazionale Calcolo Scientifico – Istituto Nazionale di Alta Matematica (GNCS-INdAM).

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] C. Dodaro, N. Leone, B. Nardi, F. Ricca, Allotment problem in travel industry: A solution based on ASP, in: RR, volume 9209 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 77–92.
- [2] C. Dodaro, G. Galatà, M. Maratea, I. Porro, An asp-based framework for operating room scheduling, *Intelligenza Artificiale* 13 (2019) 63–77.
- [3] M. Cardellini, P. D. Nardi, C. Dodaro, G. Galatà, A. Giardini, M. Maratea, I. Porro, A two-phase ASP encoding for solving rehabilitation scheduling, in: RuleML+RR, volume 12851 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 111–125.
- [4] E. Erdem, V. Patoglu, Applications of ASP in robotics, *Künstliche Intell.* 32 (2018) 143–149.
- [5] M. A. BORROTO SANTANA, K. GALLAGHER, A. IELO, I. KAREEM, F. RICCA, A. RUSSO, Question answering with llms and learning from answer sets, *Theory and Practice of Logic Programming* (2025) 1–25. doi:10.1017/S1471068425100343.
- [6] E. Erdem, M. Gelfond, N. Leone, Applications of answer set programming, *AI Magazine* 37 (2016) 53–68.
- [7] F. Riguzzi, *Foundations of Probabilistic Logic Programming Languages, Semantics, Inference and Learning*, Second Edition, River Publishers, Gistrup, Denmark, 2022.
- [8] L. D. Raedt, K. Kersting, S. Natarajan, D. Poole, Statistical relational artificial intelligence: Logic, probability, and computation, *Synthesis Lectures on Artificial Intelligence and Machine Learning* 10 (2016) 1–189.
- [9] G. Brewka, T. Eiter, M. Truszczyński, Answer set programming at a glance, *Communications of the ACM* 54 (2011) 92–103. doi:10.1145/2043174.2043195.
- [10] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Multi-shot ASP solving with clingo, *Theory and Practice of Logic Programming* 19 (2019) 27–82. doi:10.1017/S1471068418000054.
- [11] F. G. Cozman, D. D. Mauá, The joy of probabilistic answer set programming: Semantics, complexity, expressivity, inference, *International Journal of Approximate Reasoning* 125 (2020) 218–239. doi:10.1016/j.ijar.2020.07.004.
- [12] D. Azzolini, G. Mazzotta, F. Ricca, F. Riguzzi, A novel framework for reasoning over optimization problems in probabilistic answer set programming, in: M. Ortiz, R. Wassermann, T. Schaub (Eds.), *Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning, KR 2025, Melbourne, Australia, November 1-17, 2025*, 2025. URL: <https://doi.org/10.24963/kr.2025/7>. doi:10.24963/KR.2025/7.
- [13] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, T. Schaub, ASP-Core-2 input language format, *Theory and Practice of Logic Programming* 20 (2020) 294–309.
- [14] L. De Raedt, A. Kimmig, H. Toivonen, ProbLog: A probabilistic Prolog and its application in link discovery, in: M. M. Veloso (Ed.), *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, volume 7, AAAI Press, 2007, pp. 2462–2467.
- [15] W. Faber, G. Pfeifer, N. Leone, Semantics and complexity of recursive aggregates in answer set programming, *Artificial Intelligence* 175 (2011) 278–298.
- [16] G. Van den Broeck, I. Thon, M. van Otterlo, L. De Raedt, DTProbLog: A decision-theoretic probabilistic Prolog, in: M. Fox, D. Poole (Eds.), *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI Press, 2010, pp. 1217–1222.
- [17] P. Domingos, M. Richardson, Mining the network value of customers, in: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 2001, p. 57–66. doi:10.1145/502512.502525.

- [18] D. Azzolini, E. Bellodi, R. Kiesel, F. Riguzzi, Solving decision theory problems with probabilistic answer set programming, *Theory and Practice of Logic Programming* 25 (2025) 33–63. doi:10.1017/S1471068424000474.
- [19] B. Bhaisare, R. Bharati, Advancing peer review integrity: Automated reviewer assignment techniques with a focus on deep learning applications, in: A. K. Bairwa, V. Tiwari, S. K. Vishwakarma, M. Tuba, T. Ganokratanaa (Eds.), *Computation of Artificial Intelligence and Machine Learning*, Springer Nature Switzerland, Cham, 2025, pp. 312–327.
- [20] Y. Kalmukov, An algorithm for automatic assignment of reviewers to papers, *Scientometrics* 124 (2020) 1811–1850. URL: <https://doi.org/10.1007/s11192-020-03519-0>. doi:10.1007/s11192-020-03519-0.
- [21] D. Di Perro, E. Bernasconi, S. Ferilli, Automatic reviews’ assignments through answer set programming, in: *IRCDL 2025: 21st Conference on Information and Research Science Connecting to Digital and Library Science*, February 20–21 2025, Udine, Italy, 2025.
- [22] G. Amendola, C. Dodaro, N. Leone, F. Ricca, On the application of answer set programming to the conference paper assignment problem, in: *AI*IA*, volume 10037 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 164–178.
- [23] K. Topuz, T. L. Urban, R. A. Russell, M. B. Yildirim, Decision support system for appointment scheduling and overbooking under patient no-show behavior, *Annals of Operations Research* 342 (2024) 845–873. doi:10.1007/s10479-023-05799-0.
- [24] T. Cayirli, K. Yang, S. Quek, A universal appointment rule in the presence of no-shows and walk-ins, *Production and Operations Management* 21 (2012). doi:10.1111/j.1937-5956.2011.01297.x.
- [25] A. Alaeddini, K. Yang, C. Reddy, S. Yu, A probabilistic model for predicting the probability of no-show in hospital appointments, *Health care management science* 14 (2011) 146–157.
- [26] C. Dodaro, G. Galata, A. Grioni, M. Maratea, M. Mochi, I. Porro, An ASP-based solution to the chemotherapy treatment scheduling problem, *Theory and Practice of Logic Programming* 21 (2021) 835–851.
- [27] D. D. Mauá, F. G. Cozman, Complexity results for probabilistic answer set programming, *International Journal of Approximate Reasoning* 118 (2020) 133–154. doi:10.1016/j.ijar.2019.12.003.
- [28] A. Darwiche, P. Marquis, A knowledge compilation map, *Journal of Artificial Intelligence Research* 17 (2002) 229–264. doi:10.1613/jair.989.
- [29] G. Amendola, F. Ricca, M. Truszczynski, Beyond NP: quantifying over answer sets, *Theory and Practice of Logic Programming* 19 (2019) 705–721. doi:10.1017/S1471068419000140.
- [30] G. Mazzotta, F. Ricca, M. Truszczynski, Quantifying over optimum answer sets, *Theory and Practice of Logic Programming* 24 (2024) 716–736.
- [31] A. Cuteri, G. Mazzotta, F. Ricca, 2-asp(q) solving based on CEGAR, in: *AAAI*, AAAI Press, 2026, pp. 19030–19038.
- [32] D. Azzolini, N. Leone, G. Mazzotta, F. Ricca, Solving hard combinatorial optimization problems with pyqasp, in: *PADL*, volume 16401 of *Lecture Notes in Computer Science*, Springer, 2026, pp. 199–217.
- [33] D. Azzolini, G. Mazzotta, F. Ricca, F. Riguzzi, Most probable explanation in probabilistic answer set programming, in: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025*, Montreal, Canada, August 16–22, 2025, ijcai.org, 2025, pp. 9049–9057. URL: <https://doi.org/10.24963/ijcai.2025/1006>. doi:10.24963/IJCAI.2025/1006.
- [34] V. H. N. Rocha, F. Gagliardi Cozman, A credal least undefined stable semantics for probabilistic logic programs and probabilistic argumentation, in: G. Kern-Isberner, G. Lakemeyer, T. Meyer (Eds.), *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022*, 2022, pp. 309–319. doi:10.24963/kr.2022/31.
- [35] P. Totis, L. De Raedt, A. Kimmig, smProbLog: Stable model semantics in problog for probabilistic argumentation, *Theory and Practice of Logic Programming* (2023) 1–50. doi:10.1017/S147106842300008X.

- [36] J. Lee, S. Talsania, Y. Wang, Computing LPMLN using ASP and MLN solvers, *Theory and Practice of Logic Programming* 17 (2017) 942–960. doi:10.1017/S1471068417000400.
- [37] G. Brewka, J. P. Delgrande, J. Romero, T. Schaub, A general framework for preferences in answer set programming, *Artif. Intell.* 325 (2023) 104023. URL: <https://doi.org/10.1016/j.artint.2023.104023>. doi:10.1016/J.ARTINT.2023.104023.