Rushing and Strolling among Answer Sets – Navigation Made Easy
(Extended Abstract)

Johannes Klaus Fichte¹, Sarah Alice Gaggl², Dominik Rusovac²
¹TU Wien, Research Unit Database and Artificial Intelligence, Austria
²TU Dresden, Logic Programming and Argumentation Group, Germany
firstname.lastname@{tuwien.ac.at, tu-dresden.de}

Relevance to KR.
Our work presents a novel approach together with a prototypical implementation to make sense of incomprehensible solution spaces of answer set programs. As ASP has been in the interest of the KR community for quite some time, we believe that our framework paves the way for further work with interesting applications within the field of knowledge representation and reasoning. In particular, weighted faceted answer set navigation applies to configuration problems and planning problems, where it might be of interest to know to what amount certain components (partial solutions) restrict the solution space. Quantifying the effect of a navigation step involves quantitative reasoning and model counting, which may attract members of the quantitative navigation step involves quantitative reasoning and model counting community, respectively. Even more so, our approach can be used to explore tremendously large solution spaces for which enumerating all answer sets is infeasible. Dealing with a large number of disjunctive logic programs and stable model semantics (Gelfond and Lifschitz 1991) (Gelfond and Lifschitz 1988). A disjunctive logic program \( \Pi \) is a finite set of rules of the form \( \alpha_0 \mid \ldots \mid \alpha_k \leftarrow \alpha_{k+1}, \ldots, \alpha_{m+1}, \ldots, \alpha_n \) over atoms \( \alpha_i \) with \( 0 \leq k \leq m \leq n \). We denote answer sets of a program \( \Pi \) by \( \text{AS}(\Pi) \). Further, we denote brave consequences by \( \text{BC}(\Pi) := \bigcup \text{AS}(\Pi) \) and cautious consequences by \( \text{CC}(\Pi) := \bigcap \text{AS}(\Pi) \). We denote the facets of a program \( \Pi \) by \( \mathcal{F}(\Pi) := \mathcal{F}^- \cup \mathcal{F}^- \) where \( \mathcal{F}^+ := \text{BC}(\Pi) \setminus \text{CC}(\Pi) \) denotes inclusive facets and \( \mathcal{F}^- := \{ \pi \mid \alpha \in \mathcal{F}^-(\Pi) \} \) denotes exclusive facets of \( \Pi \). For further details on faceted answer set navigation, we refer to (Alrabbaa, Rudolph, and Schweizer 2018).

2 Weighted Faceted Navigation
A finite sequence \( \delta := \langle f_0, \ldots, f_n \rangle \in \Delta^\Pi \) of navigation steps towards facets \( f_i \in \mathcal{F}(\Pi) \) is called a route over \( \Pi \). The empty route is denoted by \( \epsilon \), and \( \delta \) is a subroute of \( \delta' \), denoted by \( \delta \subseteq \delta' \), whenever if \( f_i \in \delta \), then \( f_i \in \delta' \). A navigation step, denoted by \( \Pi(\delta) \), is performed by activating a facet \( f \in \{ \alpha, \neg \alpha \} \), which means that we add an integrity constraint to \( \Pi \) such that \( \Pi(\delta) = \Pi \cup \{ \leftarrow \neg \alpha \} \), if \( f = \alpha \), and otherwise, if \( f = \neg \alpha \), \( \Pi(\delta) = \Pi \cup \{ \leftarrow \alpha \} \). Thereby we land in a subset \( \text{AS}(\Pi(\delta)) \subset \text{AS}(\Pi) \) of solutions (sub-space) where each solution includes/excludes the specific inclusive/exclusive facet. To avoid stepping into empty sub-spaces we distinguish between so called safe routes denoted by \( \Delta^\Pi := \{ \delta \in \Delta^\Pi \mid \text{AS}(\Pi(\delta)) \neq \emptyset \} \) and unsafe routes \( \Delta^\Pi \setminus \Delta^\Pi \). Faceted navigation is possible as long as we are on a safe route. In case we take an unsafe route \( \delta \), we can proceed navigating toward some facet \( f \) by taking a possible redirection of \( \delta \) with respect to \( f \) from \( R(\delta, f) := \{ \delta' \subseteq \delta \mid f \in \delta', \text{AS}(\Pi(\delta')) \neq \emptyset \} \cup \{ \epsilon \} \). The so called weight \( \omega_\#(f, \Pi^\delta, \delta') \) of a facet \( f \in \mathcal{F}(\Pi) \) on route \( \delta \) and with respect to redirection \( \delta' \in R(\delta, f) \) as defined by
\[ \omega_\#(f, \Pi^\delta, \delta') := \begin{cases} \#(\Pi^\delta) - \#(\Pi^\delta'), & \text{if } (\delta, f) \notin \Delta_\#^\# \\
\#(\Pi^\delta) - \#(\Pi^\delta, f), & \text{else}. \end{cases} \]
serves the purpose of quantifying the effect of activating \( f \). It is a parameter that indicates to what a amount the activation of \( f \) restricts the solution space with respect to the
one solution, but their facet-counting weights differ. While the absolute weight $\omega_{\# AS}$ (counting answer sets) turns out to be the natural choice, counting answer sets is hard (Fichte et al. 2021b). Thus, we investigated the relative support weight $\omega_{\# F}$ (counting supported models) and the relative facet-counting weight $\omega_{\# F}$ (counting facets), which provide cheaper methods for quantifying the effect of a navigation step. 

In the original paper we prove, in a systematic comparison, which weights can be employed under the search space navigation operations and illustrate the computational complexity for computing the weights.

**Example 1.** Consider $\Pi_1 = \{a \land b \land c \land d \land e \land \neg b \land e\}$. We have $AS(\Pi_1) = \{a, e\}, \{b, c, e\}, \{b, d, e\}$ and $F(\Pi_1) = \{a, b, c, d, \pi, \overline{b}, \overline{c}, \overline{d}\}$. We discuss the facet counting weight of $\bar{c}$ is 2, the reason is that counting facets indicates to what amount solutions converge, that is, the solutions in $AS(\Pi_1) = \{a, e\}, \{b, d, e\}$ have the same absolute weight of $\bar{c}$, which can be understood as search strategies, in order to find a unique desirable solution as quick as possible. In particular, our approach applies to configuration problems and planning problems, where it might be of interest to know to what amount certain components restrict the solution space. However, weighted faceted navigation is in general useful for making sense of solution spaces, as it provides insights on the properties that certain facets of a problem have.

**Experiments.** To study the feasibility of our framework, we implemented fasb (Fichte, Gaggl, and Rusovac 2021b), an interactive prototype of our framework on top of the clingo solver (Gebser et al. 2011; Gebser et al. 2018). We conducted experiments (Fichte, Gaggl, and Rusovac 2021a) in order to (i) demonstrate the feasibility of our approach by navigating through the solution space of a PC configuration encoding with more than a billion solutions; and (ii) demonstrate that the feasibility of our approach depends on the complexity of the given program, by analyzing runtimes of navigation over two encodings of different complexity but with an identical solution space.

**3 Conclusion**

Establishing fundamental notions such as routes, navigation modes and weights of facets for faceted answer set navigation, we provide a formal and dynamic framework for navigating through the solution space of an answer set program in a systematic way. Our framework is intended as an additional layer on top of a solver, which adds functionality for exploring the search space with quantitative considerations expressed by weights of facets. Our prototypical implementation demonstrates the feasibility of our framework for an incomprehensible solution space.

**References**


