Tool Paper: A Simple and Effective ASP-Based Tool for Enumerating Minimal Hitting Sets

Mohimenul Kabir[a] and Kuldeep S Meel[b,c]

[a]National University of Singapore
[b]Georgia Institute of Technology
[c]University of Toronto

41st International Conference on Logic Programming (ICLP), Rende, Italy, 2025

Minimal Hitting Sets

- Input: A family of sets $\mathcal{S} = \{S_1, \ldots, S_k\}$
- Hitting Set: a set h such that $\forall_{S_i \in \mathcal{S}} h \cap S_i \neq \emptyset$ [VO2000]
- Minimal Hitting Set: a hitting set h of $\mathcal{S}$ when no $h' \subset h$ is a hitting set of $\mathcal{S}$

Minimal Hitting Sets

- Input: A family of sets $\mathcal{S} = \{S_1, \ldots, S_k\}$
- Hitting Set: a set h such that $\forall_{S_i \in \mathcal{S}} h \cap S_i \neq \emptyset$ [VO2000]
- Minimal Hitting Set: a hitting set h of $\mathcal{S}$ when no $h' \subset h$ is a hitting set of $\mathcal{S}$
- Example: consider $\mathcal{S} = \{\{1,2\},\{3\},\{2,3,4\}\}$.
  - The set $\{1,3\}$ hits each set of $\{\{1,2\},\{3\},\{2,3,4\}\}$. No proper subset of $\{1,3\}$ hits every set of $\mathcal{S}$.
  - The set $\{2,3\}$ hits each set of $\{\{1,2\},\{3\},\{2,3,4\}\}$. No proper subset of $\{2,3\}$ hits every set of $\mathcal{S}$.
- Applications: diagnosis [Reiter1987;AG2009], explanability [SU2006], data mining [BMR2003], and bioinformatics and computational biology [KG2004].

Minimal Hitting Sets

- Input: A family of sets $\mathcal{S} = \{S_1, \dots, S_k\}$
- Hitting Set: a set h such that $\forall_{S_i \in \mathcal{S}} h \cap S_i \neq \emptyset$ [VO2000]
- Minimal Hitting Set: a hitting set h of $\mathcal{S}$ when no $h' \subset h$ is a hitting set of $\mathcal{S}$
- Example: consider $\mathcal{S} = \{\{1,2\},\{3\},\{2,3,4\}\}$.
  - The set $\{1,3\}$ hits each set of $\{\{1,2\},\{3\},\{2,3,4\}\}$. No proper subset of $\{1,3\}$ hits every set of $\mathcal{S}$.
  - The set $\{2,3\}$ hits each set of $\{\{1,2\},\{3\},\{2,3,4\}\}$. No proper subset of $\{2,3\}$ hits every set of $\mathcal{S}$.
- Applications: diagnosis [Reiter1987;AG2009], explanability [SU2006], data mining [BMR2003], and bioinformatics and computational biology [KG2004].
- Our Goal: All minimal hitting sets (MHS) enumeration
- All MHSes Applications: Fault diagnosis [Reiter1987], system biology [VBB$^+$2013]

Answer Set Programming

- Roots in logic programming and nonmonotonic reasoning
- A rule-based language for problem encoding

$$\underbrace{h_1 \vee \ldots \vee h_\ell}_{head} \leftarrow \underbrace{b_1, \ldots, b_k, \sim b_{k+1}, \ldots, \sim b_{k+m}}_{body}.$$

Answer Set Programming

- Roots in logic programming and nonmonotonic reasoning
- A rule-based language for problem encoding

$$\underbrace{h_1 \vee \ldots \vee h_\ell}_{head} \leftarrow \underbrace{b_1, \ldots, b_k, \sim b_{k+1}, \ldots, \sim b_{k+m}.}_{body}$$

- An ASP program $P \equiv$ set of rules.
- The model of P is an *answer set* (denoted as $\mathrm{AS}(P)$).

Answer Set Programming

- Roots in logic programming and nonmonotonic reasoning
- A rule-based language for problem encoding

$$\underbrace{h_1 \vee \ldots \vee h_\ell}_{head} \leftarrow \underbrace{b_1, \ldots, b_k, \sim b_{k+1}, \ldots, \sim b_{k+m}.}_{body}$$

- An ASP program $P \equiv$ set of rules.
- The model of P is an *answer set* (denoted as $\mathrm{AS}(P)$).
- Answer set programming follows the *default negation* — everything is false unless there are some justifications.
- Consider an ASP program, $P = \{\underbrace{a \vee b \leftarrow \top.}_{r_1}\}$
  - $\{a\} \in \mathrm{AS}(P)$, since a is justified by r_1 and b is false
  - $\{b\} \in \mathrm{AS}(P)$, since b is justified by r_1 and a is false
  - $\{a, b\} \notin \mathrm{AS}(P)$, since a and b are not justified

Answer Set Programming

- Roots in logic programming and nonmonotonic reasoning
- A rule-based language for problem encoding

$$\underbrace{h_1 \vee \ldots \vee h_\ell}_{head} \leftarrow \underbrace{b_1, \ldots, b_k, \sim b_{k+1}, \ldots, \sim b_{k+m}.}_{body}$$

- An ASP program $P \equiv$ set of rules.
- The model of P is an *answer set* (denoted as $\mathrm{AS}(P)$).
- Answer set programming follows the *default negation* — everything is false unless there are some justifications.
- Consider an ASP program, $P = \{\underbrace{a \vee b \leftarrow \top.}_{r_1}\}$
  - $\{a\} \in \mathrm{AS}(P)$, since a is justified by r_1 and b is false
  - $\{b\} \in \mathrm{AS}(P)$, since b is justified by r_1 and a is false
  - $\{a, b\} \notin \mathrm{AS}(P)$, since a and b are not justified
- Main Contribution: An ASP-based MHS enumeration tool — MinHit-ASP.

From MHSes to Answer Sets

- We can compute MHSes of $\mathcal{S}$ by answer set solving
- For a family of sets $\mathcal{S}$, there is an ASP program P($\mathcal{S}$) such that AS(P($\mathcal{S}$)) correspondence to minimal hitting sets of $\mathcal{S}$

From MHSes to Answer Sets

- We can compute MHSes of $\mathcal{S}$ by answer set solving
- For a family of sets $\mathcal{S}$, there is an ASP program $\mathrm{P}(\mathcal{S})$ such that $\mathrm{AS}(\mathrm{P}(\mathcal{S}))$ correspondence to minimal hitting sets of $\mathcal{S}$

$\mathrm{P}(\mathcal{S})$ Encoding:

- for each set $S_i = \{a_1, \ldots, a_\ell\} \in \mathcal{S}$, we introduce a rule to $\mathrm{P}(\mathcal{S})$ as follows:

$$a_1 \vee \ldots \vee a_\ell \leftarrow \top.$$

From MHSes to Answer Sets

- We can compute MHSes of $\mathcal{S}$ by answer set solving
- For a family of sets $\mathcal{S}$, there is an ASP program $\mathrm{P}(\mathcal{S})$ such that $\mathrm{AS}(\mathrm{P}(\mathcal{S}))$ correspondence to minimal hitting sets of $\mathcal{S}$

$\mathrm{P}(\mathcal{S})$ Encoding:

- for each set $S_i = \{a_1, \ldots, a_\ell\} \in \mathcal{S}$, we introduce a rule to $\mathrm{P}(\mathcal{S})$ as follows:

$$a_1 \vee \ldots \vee a_\ell \leftarrow \top.$$

Correctness of $\mathrm{P}(\mathcal{S})$:

- every answer set hits every set of $\mathcal{S}$
- every answer set is minimal hitting set due to Justification

From MHSes to Answer Sets

- We can compute MHSes of $\mathcal{S}$ by answer set solving
- For a family of sets $\mathcal{S}$, there is an ASP program $\mathrm{P}(\mathcal{S})$ such that $\mathrm{AS}(\mathrm{P}(\mathcal{S}))$ correspondence to minimal hitting sets of $\mathcal{S}$

$\mathrm{P}(\mathcal{S})$ Encoding:

- for each set $S_i = \{a_1, \ldots, a_\ell\} \in \mathcal{S}$, we introduce a rule to $\mathrm{P}(\mathcal{S})$ as follows:

$$a_1 \vee \ldots \vee a_\ell \leftarrow \top.$$

Correctness of $\mathrm{P}(\mathcal{S})$:

- every answer set hits every set of $\mathcal{S}$
- every answer set is minimal hitting set due to Justification

Example: consider $\mathcal{S} = \{\{1,2\},\{3\},\{2,3,4\}\}$

- $\mathrm{P}(\mathcal{S}) = \{1 \vee 2 \leftarrow \top.\ 3 \leftarrow \top.\ 2 \vee 3 \vee 4 \leftarrow \top.\}$
- $\mathrm{P}(\mathcal{S})$ has two answer sets: $\{1,3\}$ and $\{2,3\}$, which correspond to two MHSes ($\{1,3\}$ and $\{2,3\}$) of $\mathcal{S}$.

Experimental Evaluation

Benchmark sources:

1. unique column combinations [BBF+2020]
2. cluster vertex deletion [VO2020]
3. metabolic reactions [HK2011]
4. cell signaling networks [ZS2005]
5. Connect-4 board game [DG2017]
6. frequent itemset mining [MU2013]
7. graph theory [BEG+2003]
8. ISCAS85 circuits [KNP+2009]
9. randomly generated instances [MU2013]

Experimental Evaluation

Benchmark sources:

1. unique column combinations [BBF+2020]
2. cluster vertex deletion [VO2020]
3. metabolic reactions [HK2011]
4. cell signaling networks [ZS2005]
5. Connect-4 board game [DG2017]
6. frequent itemset mining [MU2013]
7. graph theory [BEG+2003]
8. ISCAS85 circuits [KNP+2009]
9. randomly generated instances [MU2013]

Baselines:

1. Hitman [AAJ2018]
2. SHD [MU2013]
3. MtMiner [HBC2007]
4. SAT solver [ES2003;LYZ+2021]
5. Gurobi v11.0.1

Experimental Evaluation

Benchmark sources:

1. unique column combinations [BBF+2020]	2. cluster vertex deletion [VO2020]
3. metabolic reactions [HK2011]	4. cell signaling networks [ZS2005]
5. Connect-4 board game [DG2017]	6. frequent itemset mining [MU2013]
7. graph theory [BEG+2003]	8. ISCAS85 circuits [KNP+2009]
9. randomly generated instances [MU2013]	

Baselines:

1. Hitman [AAJ2018]	2. SHD [MU2013]
3. MtMiner [HBC2007]	4. SAT solver [ES2003;LYZ+2021]
5. Gurobi v11.0.1	

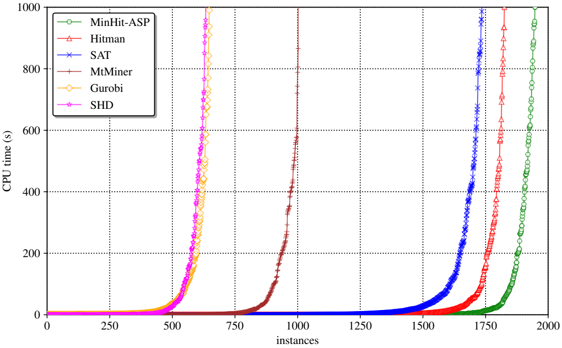
Experimental Settings:
A single core, with a time limit of 1000 seconds, a memory limit of 16 GB

Experimental Results

	SAT	ILP	MtMiner	SHD	Hitman	MinHit-ASP
#Solved	1735	647	1002	633	1846	**1948**

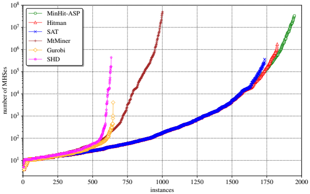
Experimental Results

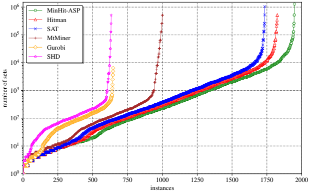
	SAT	ILP	MtMiner	SHD	Hitman	MinHit-ASP
#Solved	1735	647	1002	633	1846	**1948**
			VBS1			+MinHit-ASP
#Solved			1906			2000 (+**94**)



Strengths and Weaknesses

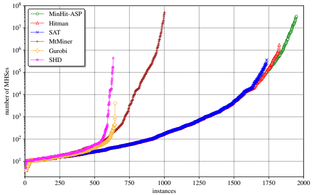
- 👍 Strengths: MinHit-ASP is competitive with the existing MHS solvers in terms of the number of MHSes enumerated and size of the instances.

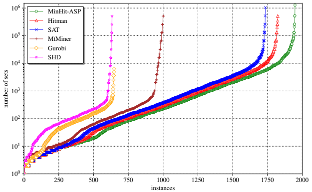




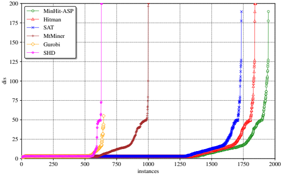
Strengths and Weaknesses

- Strengths: MinHit-ASP is competitive with the existing MHS solvers in terms of the number of MHSes enumerated and size of the instances.





- Weaknesses: MinHit-ASP struggles with instances with larger size or higher disjunctions



Conclusion

- We introduce MinHit-ASP, an ASP-based tool for MHSes enumeration
- Our tool supports versatile features: *brave* and *cautious* reasoning, *preference* among MHSes, *counting* MHSes, and others.
- MinHit-ASP outperforms existing MHS solvers in enumerating all MHSes

`https://zenodo.org/records/15104697`