

# 1 Explanation Generation

---

**Algorithm 1:** EXPLAINCSP( $\mathcal{C}, U, f, I$ )

---

**Input** :  $\mathcal{C}$ , a CNF  $\mathcal{C}$  over a vocabulary  $V$   
**Input** :  $U$ , a user vocabulary  $U \subseteq V$   
**Input** :  $f$ , a cost function  $f : 2^{lits(U)} \rightarrow \mathbb{N}$ .  
**Input** :  $I$ , a partial interpretation over  $U$   
**Output** :  $E$ , a sequence of explanation steps as implications  $I_{expl} \implies N_{expl}$

```

1 SAT  $\leftarrow$  INITSAT( $\mathcal{C}$ )
2  $E \leftarrow \langle \rangle$ 
3  $I_{end} \leftarrow$  OPTIMALPROPAGATE( $U, I$ )
4 while  $I \neq I_{end}$  do
5    $X \leftarrow$  BESTSTEP( $\mathcal{C}, f, I_{end}, I$ )
6    $I_{best} \leftarrow I \cap X$ 
7    $N_{best} \leftarrow$  OPTIMALPROPAGATE( $U, I_{best}$ )  $\setminus I$ 
8   add  $\{I_{best} \implies N_{best}\}$  to  $E$ 
9    $I \leftarrow I \cup N_{best}$ 
10 end
11 return  $E$ 

```

---



---

**Algorithm 2:** OPTIMALPROPAGATE( $\mathcal{U}, I, \text{SAT}$ )

---

**Input** :  $U$ , a user vocabulary  $U \subseteq V$   
**Optional:**  $I$ , a partial interpretation over  $U$ .  
**State** : SAT, a SAT solver initialised with a CNF.  
**Output** : The projection onto  $U$  of the intersection more precise than  $I$ .

```

1  $sat?, \mu \leftarrow$  SAT.solve( $I$ )
2  $\mu \leftarrow \{l \mid \text{var}(l) \in U\}$ 
3  $b \leftarrow$  a new blocking variable
4 while true do
5   SAT.addClause( $\neg b_i \bigvee_{l \in \mu} \neg l$ )
6    $sat?, \mu' \leftarrow$  SAT.solve( $I \wedge \{b_i\}$ )
7   if not sat? then
8     SAT.addClause( $\neg b_i$ )
9     return  $\mu$ 
10  end
11   $\mu \leftarrow \mu \cap \mu'$ 
12 end

```

---



---

**Algorithm 3:** BESTSTEP-C-OUS( $f, I_{end}, I, \text{SAT}$ )

---

**Input** :  $f$ , a cost function  $f : 2^{\mathcal{G}} \rightarrow \mathbb{N}$  over CNF  $\mathcal{G}$ .  
**Input** :  $I_{end}$ , the cautious consequence, the set of literals that hold in all models.  
**Input** :  $I$ , a partial interpretation s.t.  $I \subseteq I_{end}$ .  
**State** : SAT, a SAT solver initialised with a CNF.  
**Output** : a single best explanation step

```

1  $\mathcal{A} \leftarrow I \cup (\overline{I_{end}} \setminus \bar{I})$  // Optimal US is subset of A
2 set  $p \triangleq \sum_{l \in \overline{I_{end}}} l = 1$  i.e. exactly one of  $\overline{I_{end}}$  in the unsatisfiable subset.
3 return C-OUS( $f, p, \mathcal{A}$ )

```

---

---

**Algorithm 4:** c-OUS( $f, p, \mathcal{A}, \text{SAT}$ )

---

**Input** :  $f$ , a cost function  $f : 2^{\mathcal{A}} \rightarrow \mathbb{N}..$   
**Input** :  $p$ , a predicate  $p : 2^{\mathcal{A}} \rightarrow \{t, f\}..$   
**Input** :  $A$ , a set of assumption literals, s.t.  $\mathcal{C} \cup A$  is unsatisfiable.  
**State** :  $\text{SAT}$ , a SAT solver initialised with a CNF.  
**Output** :  $(p, f) - \text{OUS}$ , a subset  $\mathcal{A}'$  of  $\mathcal{A}$  that satisfies  $p$  s.t.  $\mathcal{C} \cup \mathcal{A}'$  is UNSAT and  $\mathcal{A}'$  is  $f$ -optimal.

```
1  $\mathcal{H} \leftarrow \emptyset$ 
2 while true do
3    $\mathcal{A}' \leftarrow \text{CONDOPTHITTINGSET}(f, p, \mathcal{A}, \mathcal{H})$ 
4   if  $\neg \text{SAT}(\mathcal{A}')$  then
5     return  $\mathcal{A}'$ 
6   end
7    $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{A} \setminus \mathcal{A}'\}$ 
8 end
```

---

## 2 MIP model

We assume  $U \subseteq V$  a set of user variables  $U$  defined over a vocabulary  $V$  of the CNF  $\mathcal{C}$ . Given an initial assignment  $I$ , where  $\text{vars}(I) \subseteq U$ ,  $I_{\text{end}}$  is as the cautious consequence (the set of literals that hold in all models) of  $\mathcal{C}$ .

The Mixed Integer Programming model for computing c-OUSes has many similarities with a set covering problem. The CONDOPTHITTINGSET computes the optimal hitting set over a collection of sets  $\mathcal{H}$ , where optimal means "among those satisfying  $p$ ".

In practice, to ensure that MIP model takes advantage of the incrementality of the problem, namely across different c-OUS calls, the specification is defined on the full set of literals of  $I_{\text{end}}$ . The constrained optimal hitting set is described by

- $x_l = \{0, 1\}$  is a boolean decision variable if the literal is selected or not.
- $w_l = f(l)$  is the cost assigned to having the literal in the hitting set ( $\infty$  otherwise).
- $c_{lj} = \{0, 1\}$  is 1 (0) if the literal  $l$  is (not) present in hitting set  $j$ .

$$\min_x \sum_{l \in I_{\text{end}} \cup \overline{I_{\text{end}}}} w_l \cdot x_l \quad (1)$$

$$\sum_{l \in I_{\text{end}} \cup \overline{I_{\text{end}}}} x_l \cdot c_{lj} \geq 1, \forall j \in \{1..|hs|\} \quad (2)$$

The basic specification finds the optimal hitting set

$$\sum_{l \in \overline{I_{\text{end}} \setminus \bar{I}}} x_l \geq 1, \forall j \in \{1..|hs|\} \quad (3)$$