

1 Explanation Generation

Algorithm 1: CSP-Explain(\mathcal{T} , f [, \mathcal{I}_0])

```

input           :  $\mathcal{T}$  set of constraints
input           :  $f$  a consistent objective function
optional input:  $\mathcal{I}_0$  a partial interpretation
output          : Explanation sequence
1 begin
2    $\mathcal{I} \leftarrow \mathcal{I}_0$  // Initial partial interpretation
3    $\mathcal{I}_{end} \leftarrow \text{propagate}(\mathcal{I}, \mathcal{T})$  // Goal state
4    $Seq \leftarrow \text{empty set}$  // explanation sequence
5   while  $\mathcal{I} \neq \mathcal{I}_{end}$  do
6      $\mathcal{F} \leftarrow \mathcal{I}_{end} \setminus \mathcal{I}$ ; // Facts to be derived
7      $\mathcal{F}' \leftarrow \{\neg \mathcal{F}\}$ ; // Set with all negated literals of  $\mathcal{F}$ 
8     //
9     // Propagate all negated literals OMUS finds the smallest explanation
10    //
11     $X \leftarrow \text{OMUS}(\mathcal{F}' \wedge \mathcal{I} \wedge \mathcal{S})$ ;
12     $E \leftarrow \mathcal{I} \cap X$ ; // Explanation used
13     $\mathcal{N} \leftarrow \text{propagate}(E \wedge \mathcal{S})$ ; // Newly derived facts
14     $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{N}$ ; // Update known facts
15    //
16    // Add explanation for newly derived facts to explanation sequence
17    //
18    for  $n \in \mathcal{N}$  do
19      |  $(E_n, \mathcal{S}_n, n)$  to  $Seq$ ;
20    end
21  end
22 end

```

2 OMUS Algorithm

Note that if we assign a unit weight to every element in the subset, we reduce the problem of finding an OMUS to finding a SMUS.

Definition 1. Let Γ be a collection of sets and $HS(\Gamma)$ the set of all hitting sets on Γ and let f be an valid objective function. Then a hitting set $h \in \Gamma$ is said to be an **optimal** hitting set if $\forall h' \in HS(\Gamma)$ we have that $f(h) \leq f(h')$.

Property 1. The **optimal** hitting set of a collection of sets Γ is denoted by $OHS(\Gamma)$.

The algorithm is based on the following observation:

Proposition 1. A set $\mathcal{U} \subseteq \mathcal{F}$ is an OMUS of \mathcal{F} if and only if \mathcal{U} is an optimal hitting set of $MCSes(\mathcal{F})$

Lemma 1. Let $\mathcal{K} \subseteq MCSes(\mathcal{F})$. Then a subset \mathcal{U} of \mathcal{F} is an OMUS if \mathcal{U} is a optimal hitting set on \mathcal{K} and \mathcal{U} is unsatisfiable

Algorithm 2: OMUS-Delayed(\mathcal{F} , $[f, \mathcal{H}_0]$)

```
input      :  $\mathcal{F}$  a CNF formula
input      :  $cost$  a cost function
optional input:  $\mathcal{H}_0$  initial collection of disjoint Minimum Correction Sets
output     : OMUS( $\mathcal{F}$ )
1 begin
2    $\mathcal{H} \leftarrow \text{DisjointMCS}(\mathcal{F})$ 
3   while true do
4      $hs \leftarrow \text{OptimalHittingSet}(\mathcal{H}, cost)$  // Find optimal solution
5      $\mathcal{F}' \leftarrow \{c_i | e_i \in hs\}$ 
6      $(sat?, \kappa) \leftarrow \text{SatSolver}(\mathcal{F}')$ 
7     // If SAT,  $\kappa$  contains the satisfying truth assignment
8     // IF UNSAT,  $\mathcal{F}'$  is the OMUS
9     if not sat? then
10      break
11       $\mathcal{C} \leftarrow \mathcal{F} \setminus \text{Grow}(\mathcal{F}')$ 
12       $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{C}\}$ 
13       $nonOptLevel \leftarrow 0$ 
14      // Find a series of non-optimal solutions
15      while true do
16        switch nonOptLevel do
17          case 0
18            // adds clause to current core to the current hitting set
19             $hs \leftarrow \text{FindIncrementalHittingSet}(\mathcal{H}, \mathcal{F}', hs)$ 
20          case 1
21             $hs \leftarrow \text{FindGreedyHittingSet}(\mathcal{H})$ 
22          end
23           $\mathcal{F}' \leftarrow \{c_i | e_i \in hs\}$ 
24           $(sat?, \kappa) \leftarrow \text{SatSolver}(\mathcal{F}')$ 
25          if not sat? then
26            switch nonOptLevel do
27              case 0
28                 $nonOptLevel \leftarrow 1$ 
29              case 1
30                break
31            end
32          else
33             $\mathcal{C} \leftarrow \mathcal{F} \setminus \text{Grow}(\mathcal{F}')$ 
34             $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{C}\}$ 
35             $nonOptLevel \leftarrow 0$ 
36          end
37        end
38      end
39    end
40    return  $(\mathcal{F}', cost(\mathcal{F}'))$ 
41 end
```

3 MIP hitting set problem specification

For the set of clauses $\mathcal{C} = \{c_1, \dots, c_{|\mathcal{C}|}\}$ with weights $\mathcal{W} = \{w_1, \dots, w_{|\mathcal{C}|}\}$ in the collection of sets \mathcal{H} . For Example:

$$\begin{aligned}\mathcal{C} &= \{c_1, \dots, c_6\} \\ \mathcal{W} &= \{w_1 = 20, w_2 = 20, w_3 = 10, w_4 = 10, w_5 = 10, w_6 = 20\} \\ \mathcal{H} &= \{c_3\}, \{c_2, c_4\}, \{c_1, c_4\}, \{c_1, c_5, c_6\}\end{aligned}\tag{1}$$

The optimal hitting set can be formulated as an integer linear program.

$$\min \sum_{i \in \{1..|\mathcal{C}|\}} w_i \cdot x_i \tag{2}$$

$$\sum_{i \in \{1..|\mathcal{C}|\}} x_i \cdot h_{ij} \geq 1, \forall j \in \{1..|\mathcal{H}|\} \tag{3}$$

$$x_i = \{0, 1\} \tag{4}$$

- w_i is the input cost/weight associated with clause i in
- x_i is a boolean decision variable if constraint/clause c_i is chosen or not.
- Equation 3, h_{ij} is a boolean input variable corresponding to if constraint/clause i is in set to hit j .

4 Future Work

References