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 $\mathcal{I}_{end} \leftarrow \texttt{propagate}(\mathcal{I}_0,\ \mathcal{T})$ `// Goal state`
3 $\mathcal{I} \leftarrow \mathcal{I}_0$ `// Initial partial interpretation`
4 $Seq \leftarrow empty\ set$ `// explanation sequence`
5   **while** $\mathcal{I} \neq \mathcal{I}_{end}$ **do**
6   **for** $i \in \mathcal{I}_{end} \setminus \mathcal{I}$ **do**
7 $X_i \leftarrow \texttt{OMUS}(\{\neg i\} \wedge \mathcal{I} \wedge \mathcal{S})$
8 $E_i \leftarrow \mathcal{I} \cap X_i$ `// Facts used`
9 $S_i \leftarrow \mathcal{T} \cap X_i$ `// Constraint used`
10 $\mathcal{N}_i \leftarrow \texttt{propagate}(E_i \wedge \mathcal{S}_i)$ `// Newly derived facts`
11   **end**
12 $(E_{best}, S_{best}, N_{best}) \leftarrow (E_i, S_i, N_i)$ with lowest $f(E_i, S_i, N_i)$
13   append $(E_{best}, S_{best}, N_{best})$ to Seq
14 $\mathcal{I} \leftarrow \mathcal{I} \cup \{N_{best}\}$
15   **end**
16 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(Γ) 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   : *$\mathcal{OMUS}(\mathcal{F})$*

1 begin

2 $\mathcal{H} \leftarrow$ `DisjointMCS`$(\mathcal{F})$

3   **while** *true* **do**

4 $hs \leftarrow$ `OptimalHittingSet`$(\mathcal{H}, cost)$ `// Find optimal solution`

5 $(sat?, \mu) \leftarrow$ `SatSolver`(hs)
              `// If SAT, `μ` contains the satisfying truth assignment`
              `// IF UNSAT, `hs` is the OMUS`

6   **if** *not sat?* **then**

7   **break**

8   **end**

9 $\mathcal{C} \leftarrow \mathcal{F} \setminus$ `Grow`(hs)

10 $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{C}\}$

11 $nonOptLevel \leftarrow 0$
        `// Find a series of non-optimal solutions`

12   **while** *true* **do**

13   **switch** *nonOptLevel* **do**

14   **case** *0*
                    `// Add/Remove clause (choose clause appears most frequently in the`
                        `set of hitting sets so far)`

15 $hs \leftarrow$ `FindIncrementalHittingSet`$(H, \mathcal{C}, hs)$

16   **case** *1*
                    `// Greedy algorithm`
                    `// 'Approximation algorithms for combinatorial problems' (1973)`

17 $hs \leftarrow$ `FindGreedyHittingSet`$(\mathcal{H})$

18   **end**

19 $(sat?, \mu) \leftarrow$ `SatSolver`(hs)

20   **if** *not sat?* **then**

21   **switch** *nonOptLevel* **do**

22   **case** *0*

23 $nonOptLevel \leftarrow 1$

24   **case** *1*

25   **break**

26   **end**

27   **else**

28 $\mathcal{C} \leftarrow \mathcal{F} \setminus$ `Grow`(hs)

29 $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{C}\}$

30 $nonOptLevel \leftarrow 0$

31   **end**

32   **end**

33   **return** $(hs', \ cost(hs))$

34 end

3 MIP hitting set problem specification

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

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

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

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

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

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

- 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