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*

begin
  $\mathcal{I}_{end} \leftarrow$ `propagate(`$\mathcal{I}_0$`, `$\mathcal{T}$`)` `// Goal state`
  $\mathcal{I} \leftarrow \mathcal{I}_0$ `// Initial partial interpretation`
  $Seq \leftarrow$ *empty set* `// explanation sequence`
  while $\mathcal{I} \neq \mathcal{I}_{end}$ **do**
    for $i \in \mathcal{I}_{end} \setminus \mathcal{I}$ **do**
      $X_i \leftarrow$ `OMUS(`$\{\neg i\} \wedge \mathcal{I} \wedge \mathcal{S}$`)`
      $E_i \leftarrow \mathcal{I} \cap X_i$ `// Facts used`
      $S_i \leftarrow \mathcal{T} \cap X_i$ `// Constraint used`
      $\mathcal{N}_i \leftarrow$ `propagate(`$E_i \wedge \mathcal{S}_i$`)` `// Newly derived facts`
    end
    $(E_{best}, S_{best}, N_{best}) \leftarrow (E_i, S_i, N_i)$ with lowest $f(E_i, S_i, N_i)$
    append $(E_{best}, S_{best}, N_{best})$ to Seq
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{N_{best}\}$
  end
end

Algorithm 2: CSP-Explain-Incremental($\mathcal{T},\ f\ [,\ \mathcal{I}_0]$)

$\mathcal{I}_{end} \leftarrow$ PROPAGATE($\mathcal{I}_0,\ \mathcal{T}$)
$\mathcal{I} \leftarrow \mathcal{I}_0$
$Seq \leftarrow \emptyset$
$\mathcal{M} \leftarrow \emptyset$
while $\mathcal{I} \neq \mathcal{I}_{end}$ **do**
  for $i \in \mathcal{I}_{end} \setminus \mathcal{I}$ **do**
    $X_i,\ \mathcal{MSS} \leftarrow$ OMUS-DELAYED($\{\neg i\} \wedge \mathcal{I} \wedge \mathcal{S},\ \mathcal{M}$)
    $E_i \leftarrow \mathcal{I} \cap X_i$
    $S_i \leftarrow \mathcal{T} \cap X_i$
    $\mathcal{N}_i \leftarrow$ PROPAGATE($E_i \wedge \mathcal{S}_i$)
    $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{MSS}$
  end
  $(E_{best}, S_{best}, N_{best}) \leftarrow (E_i, S_i, N_i)$ with lowest $f(E_i, S_i, N_i)$
  append $(E_{best}, S_{best}, N_{best})$ to Seq
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{N_{best}\}$
end

2 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:

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

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

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

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

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

- 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.

Algorithm 3: OMUS-DELAYED($\mathcal{F}, f_{cost}$)

```
   // grow mss from input mss
 1 K ← ∅
 2 foreach MSS ∈ M do
 3 |  MSS' ← GROW(F ∩ MSS)
 4 |  K ← K ∪ {F \ MSS'}
 5 end
 6 mode ← mode_greedy
 7 while true do
 8 |  while true do
 9 |  |  switch nonOptLevel do
10 |  |  |  case mode_incr
11 |  |  |  |  hs ← FINDINCREMENTALHITTINGSET(K, C, hs)
12 |  |  |  case mode_greedy
13 |  |  |  |  hs ← FINDGREEDYHITTINGSET(K)
14 |  |  end
15 |  |  (sat?, κ) ← SATSOLVER(hs)
16 |  |  if not sat? then
17 |  |  |  switch nonOptLevel do
18 |  |  |  |  case mode_incr
19 |  |  |  |  |  mode ← mode_greedy
20 |  |  |  |  case mode_greedy
21 |  |  |  |  |  mode ← mode_opt
22 |  |  |  |  |  break
23 |  |  |  end
24 |  |  else
25 |  |  |  MSS ← GROW(hs)
26 |  |  |  K ← K ∪ {F \ MSS}
27 |  |  |  M ← M ∪ {MSS}
28 |  |  |  mode ← mode_incr
29 |  end
30 |  hs ← OPTIMALHITTINGSET(K, f_cost)
31 |  (sat?, κ) ← SATSOLVER(hs)
32 |  if not sat? then
33 |  |  return hs, M
34 |  end
35 |  MSS ← GROW(hs)
36 |  K ← K ∪ {F \ MSS}
37 |  M ← M ∪ {MSS}
38 |  mode ← mode_incr
39 end
```

3 Future Work

References