Proof Logging for McSplit

Ciaran McCreesh James Trimble

University of Glasgow

Proof Logging for Maximum Common Subgraph

- If the solver ever outputs the wrong answer, we want to be able to detect this.
  - Can also detect if we get the right answer but by spurious reasoning.
- Compile a problem instance into a pseudo-Boolean (OPB) optimisation instance.
- Provide a machine-verifiable cutting planes proof of optimality.
  - All of the reasoning McSplit does *could* in theory be carried out by a sufficiently clever PB solver.

A Pseudo-Boolean Encoding

Variables define an injective partial mapping:

$$x_{f,s} \in \{0,1\} \qquad f \in \mathrm{V}(F),\ s \in \mathrm{V}(S) \cup \{\bot\}$$

$$x_{f,\bot} + \sum_{s \in \mathrm{V}(S)} x_{f,s} = 1 \qquad f \in \mathrm{V}(F)$$

$$\sum_{f \in \mathrm{V}(F)} x_{f,s} \leq 1 \qquad s \in \mathrm{V}(S)$$

A Pseudo-Boolean Encoding

Adjacency:

$$\overline{x}_{f,s} + x_{g,\bot} + \sum_{t \in \mathsf{N}(s)} x_{g,t} \geq 1 \qquad f \in \mathsf{V}(F),\ g \in \mathsf{N}(f),\ s \in \mathsf{V}(S)$$

$$\overline{x}_{f,s} + x_{g,\bot} + \sum_{t \in \overline{\mathsf{N}}(s)} x_{g,t} \geq 1 \qquad f \in \mathsf{V}(F),\ g \in \overline{\mathsf{N}}(f),\ s \in \mathsf{V}(S)$$

A Pseudo-Boolean Encoding

Objective:

$$\text{maximise} \quad \sum_{f \in \mathrm{V}(F)} \sum_{s \in \mathrm{V}(S)} x_{f,s}$$

What This Looks Like

```
* #variable= 110 #constraint= 930
min: -1 x0_0 -1 x0_1 -1 x0_2 ... -1 x9_8 -1 x9_9 ;
* vertex 0 domain
1 x0_0 1 x0_1 1 x0_2 ... x0_9 1 x0_null >= 1 ;
-1 x0_0 -1 x0_1 -1 x0_2 ... x0_null >= -1 ;
...
* injectivity on value 0
-1 x0_0 -1 x1_0 -1 x2_0 ... -1 x9_0 >= -1 ;
...
* adjacency 0 maps to 0
1 ~x0_0 ... 1 x1_5 1 x1_8 1 x1_9 1 x1_null >= 1 ;
1 ~x0_0 ... 1 x2_5 1 x2_8 1 x2_9 1 x2_null >= 1 ;
...
```

Proof Logging for Backtracking Search

- Log new incumbents as we find them.
- Idea: every time we backtrack, assert that the trail "obviously" implies a contradiction.
- "Obviously" means that asserting the trail, together with every constraint known so far, leads to contradiction via unit propagation.
    - PB unit propagation is integer bounds consistency, and so can do more than SAT unit propagation.
- We can then add the negation of the trail as a new constraint.

Unit Propagation Isn't Strong Enough

- Unit propagation captures everything McSplit does with adjacency propagation, and the use of CP-style variables.
- It does *not* capture the all-different bound calculation.
- We have to manually derive new constraints using cutting planes rules to help out.

The Bounds Calculation

- The only problematic case is when a "first" partition has fewer values than its corresponding "second" partition.
- In CP terms: we have a set of $|F_n|$ variables that can take at most $|S_n|$ non-$\bot$ values between them. This is known as a *Hall violator*.
- We know how to deal with this in general (see AAAI 2020). This particular case is especially easy because of the partitioning structure.
    - Sum up the "at least one value" constraints for each offending "first" partition.
    - Sum up the "injectivity" constraints for each offending "second" partition.
    - Add all of this to the objective constraint.

What This Looks Like

```
pseudo-Boolean proof version 1.0
f 930 0
o x0_0 x1_1 x2_4 x3_5 x6_6 x8_2
u 1 ~x0_0 1 ~x6_6 1 ~x3_5 1 ~x8_2 1 ~x1_1 1 ~x2_4 >
u 1 ~x0_0 1 ~x6_6 1 ~x3_5 1 ~x8_2 1 ~x1_1 >= 1 ;
p 5 15 + 22 + 933 + 937 +
u 1 ~x0_0 1 ~x6_6 1 ~x3_5 1 ~x8_2 1 ~x1_4 >= 1 ;
p 5 15 + 19 + 22 + 25 + 933 + 940 +
u 1 ~x0_0 1 ~x6_6 1 ~x3_5 1 ~x8_2 >= 1 ;
...
u >= 1 ;
c 1202 0
```

Running It

```
$ mcsp min_max \
    mcs-instances/mcs30_r01_s15.{A,B}00 \
    -o proof.opb -p proof.log
Solution size 11
Nodes:                        5014
CPU time (ms):                58
$ ls -lh proof.{opb,log}
548K Jun 21 19:20 proof.log
311K Jun 21 19:20 proof.opb
$ veripb proof.opb proof.log
INFO:root:total time: 0.53s
Verification succeeded.
```

Connectedness

- The PB encoding is a bit awkward…
- Surprisingly, no further help is needed for unit propagation.

http://www.dcs.gla.ac.uk/~ciaran
ciaran.mccreesh@glasgow.ac.uk

University of Glasgow