Proof logging for RoundingSat

Marc Vinyals

Waipapa Taumata Rau – University of Auckland

with Wietze Koops, Jakob Nordström, Andy Oertel

RoundingSat

- Native pseudoBoolean reasoning

Decision problems

- Conflict-driven learning
- Linear programming

Optimisation problems

- Linear
- Core-guided

Conflict analysis

- Conflict: $Ax \geq a$
- Reason: $Bx \geq b$

- Weaken reason
- Divide
- Add to conflict

Example

```
pol 1 2 ~x1 3 * + x2 + 3 / 2 * +
```

Weaken away $3x_1$ and $\neg x_2$ from C_2, divide by 3, multiply by 2, and add to C_1

Fractional Infeasibility

- Infeasible LP

- Extract Farkas multipliers
- Add

Example

```
pol 1 * 4 2 * 2 + 3 * 5 +
```

Add C_1, C_2, C_3 with multipliers $4, 2, 5$

MIR Cuts

- Added if feasible LP but not IP
- Create linear combination
- Apply MIR cut

- VeriPB does not support MIR natively
- Simulate via redundance rule
- Not implemented in RoundingSat

Solution Improving Optimisation

- Find solution α with $f(\alpha) = u$
- Objective improving constraint $f < u$

Example

```
soli x1 x2 ~x3
```

Found solution $x_1 = 1, x_2 = 1, x_3 = 0$

Solution Improving Optimisation

- Find solution α with $f(\alpha) = u$
- Objective improving constraint $f < u$

Example

```
soli x1 x2 ~x3
```

Found solution $x_1 = 1, x_2 = 1, x_3 = 0$

- Find contradiction
- Last solution was optimal
- Standard logging

Core-guided Optimisation: Core Extraction

A sloppy take on *Certified Core-Guided MaxSAT Solving*

- Core: constraint falsified by assumptions

- Obtain through conflict analysis
- Standard logging

- Round to cardinality
- Weaken and divide

Core-guided Optimisation: Objective Reformulation

Notation: $X_S = \sum_{i \in S} x_i$

- Have core lower bound $X_S \geq d$

Conceptually

- Add new variables $y_b \leftrightarrow [\![X_S \geq b]\!]$
- Reformulate objective to $f_r = f_o + w\Delta_K$ where $\Delta_K = Y_{>d} - X_S + d$
- Observe that $\Delta_K = 0$, hence $f_o = f_r$

Core-guided Optimisation: Objective Reformulation

Notation: $X_S = \sum_{i \in S} x_i$

- Have core lower bound $X_S \geq d$

Conceptually

- Add new variables $y_b \leftrightarrow [\![X_S \geq b]\!]$
- Reformulate objective to $f_r = f_o + w\Delta_K$ where $\Delta_K = Y_{>d} - X_S + d$
- Observe that $\Delta_K = 0$, hence $f_o = f_r$

In Practice

- Add variables lazily, starting with y_{d+1}
- $\Delta_K^{(i)} = Y_{[d+1,d+i]} - X_S + d$
- Observe that $\Delta_K^{(i)} \leq 0$, hence $f_o \geq f_r$
- Global reformulation constraint $\Delta = \sum_K w_K \Delta_K^{(i_K)} \leq 0$
  (We'll see how to log Δ later)

Core-guided Optimisation: Core Upper Bound

- Trivial core upper bound $X_S \leq |S|$
- Better(?) bound from objective improving constraint

Core-guided Optimisation: Core Upper Bound

- Trivial core upper bound $X_S \leq |S|$
- Better(?) bound from objective improving constraint

- Suppose $f_o \leq u$
- Derive $f_r := f_o + \Delta \leq u$
- Weaken variables outside S
- Round to a cardinality constraint
- Upper bound on the core of the form $X_S \leq u'$
- Standard logging

Core-guided Optimisation: Reification: At Least

- Semantics: $[\![X_S \geq d+1]\!] \rightarrow y_{d+1}$
- LTF: $R^{(1)\leftarrow} := \xi y_{d+1} \geq X_S - d$, with ξ large enough
- $\xi = \min(u', |S|) - d$

Core-guided Optimisation: Reification: At Least

> **Redundance rule:**
> Infer C if we can prove
> $F \wedge \neg C \models (F \wedge C)\restriction_{\omega}$
> for some substitution ω

- Semantics: $[\![X_S \geq d+1]\!] \rightarrow y_{d+1}$
- LTF: $R^{(1)\leftarrow} := \xi y_{d+1} \geq X_S - d$, with ξ large enough
- $\xi = \min(u', |S|) - d$

- Log $\xi y_{d+1} - X_S \geq -d$ with redundance rule and substitution $y_{d+1} = 1$

Example

```
red +2  y2 +1 ~x1 +1 ~x2 +1 ~x3 >= 2 ; y2 -> 1
```

Core-guided Optimisation: Reification: At Least

> Redundance rule:
> Infer C if we can prove
> $F \wedge \neg C \vDash (F \wedge C)\restriction_\omega$
> for some substitution ω

- Semantics: $[\![X_S \geq d+1]\!] \rightarrow y_{d+1}$
- LTF: $R^{(1)\leftarrow} := \xi y_{d+1} \geq X_S - d$, with ξ large enough
- $\xi = \min(u', |S|) - d$

- Log $\xi y_{d+1} - X_S \geq -d$ with redundance rule and substitution $y_{d+1} = 1$

Example

```
red +2  y2 +1 ~x1 +1 ~x2 +1 ~x3 >= 2 ; y2 -> 1
```

- Constraints already in the DB syntactically untouched by substitution do not need proving
- $R^{(1)\leftarrow}\restriction_{y_{d+1}=1}$ becomes $-X_S \geq -d - \xi$
  - Trivial for $\xi = |S| - d$
  - Syntactically identical to the core upper bound otherwise

Core-guided Optimisation: Reification: At Most

- **Semantics:** $y_{d+1} \rightarrow [\![X_S \geq d+1]\!]$
- **LTF:** $y_{d+1} \leq X_S - d$

Core-guided Optimisation: Reification: At Most

> **Redundance rule:**
> Infer C if we can prove
> $F \wedge \neg C \models (F \wedge C)\restriction_{\omega}$
> for some substitution ω

- Semantics: $y_{d+1} \rightarrow [\![X_S \geq d+1]\!]$
- LTF: $y_{d+1} \leq X_S - d$
- Log $R^{(1)\rightarrow} := X_S - y_{d+1} \geq d$ with redundance rule and substitution $y_{d+1} = 0$

Example

```
red +2  y2 +1 ~x1 +1 ~x2 +1 ~x3 >= 2 ; y2 -> 1
red +1 ~y2 +1  x1 +1  x2 +1  x3 >= 2 ; y2 -> 0
```

Core-guided Optimisation: Reification: At Most

> Redundance rule:
> Infer C if we can prove
> $F \wedge \neg C \vDash (F \wedge C)\restriction_\omega$
> for some substitution ω

- Semantics: $y_{d+1} \to [\![X_S \geq d+1]\!]$
- LTF: $y_{d+1} \leq X_S - d$
- Log $R^{(1)\to} := X_S - y_{d+1} \geq d$ with redundance rule and substitution $y_{d+1} = 0$

Example

```
red +2  y2 +1 ~x1 +1 ~x2 +1 ~x3 >= 2 ; y2 -> 1
red +1 ~y2 +1  x1 +1  x2 +1  x3 >= 2 ; y2 -> 0
```

- Constraints in the DB still untouched
- $R^{(1)\leftarrow}\restriction_{y_{d+1}=0}$ becomes $-X_S \geq -d$
  - implied by $\neg R^{(1)\to} := -X_S + y_{d+1} \geq -d + 1$ (weaken y_{d+1})
- $R^{(1)\to}\restriction_{y_{d+1}=0}$ becomes $X_S \geq d$
  - syntactically identical to the core lower bound

Core-guided Optimisation: Reification: At Most

Redundance rule:
Infer C if we can prove
$F \wedge \neg C \models (F \wedge C)\restriction_\omega$
for some substitution ω

- Semantics: $y_{d+1} \rightarrow [\![X_S \geq d+1]\!]$
- LTF: $y_{d+1} \leq X_S - d$
- Log $R^{(1)\rightarrow} := X_S - y_{d+1} \geq d$ with redundance rule and substitution $y_{d+1} = 0$

Example

```
red +2  y2 +1 ~x1 +1 ~x2 +1 ~x3 >= 2 ; y2 -> 1
red +1 ~y2 +1  x1 +1  x2 +1  x3 >= 2 ; y2 -> 0
```

- Constraints in the DB still untouched
- $R^{(1)\leftarrow}\restriction_{y_{d+1}=0}$ becomes $-X_S \geq -d$
  - implied by $\neg R^{(1)\rightarrow} := -X_S + y_{d+1} \geq -d + 1$ (weaken y_{d+1})
- $R^{(1)\rightarrow}\restriction_{y_{d+1}=0}$ becomes $X_S \geq d$
  - syntactically identical to the core lower bound
- $\Delta^{(1)}$ syntactically identical to $R^{(1)\rightarrow}$

Core-guided Optimisation: Lazy Variable Expansion

- Log $y_{d+i} \leftrightarrow [\![X_S \geq d+i]\!]$ with redundance rule

- $\Delta^{(i)} = \left(\Delta^{(i-1)} + (i-1)R^{(i)}\right)/i$
- Rebuild global Δ from scratch
- Smarter data structure? Likely overkill

- Update core upper bounds?

Core-guided Optimisation: Hardening

- Suppose wx appears in the (reformulated) objective
- Suppose $u \geq f_o \geq f_r \geq l$
- If $w + l > u$, then $x = 0$

Core-guided Optimisation: Hardening

- Suppose wx appears in the (reformulated) objective
- Suppose $u \geq f_o \geq f_r \geq l$
- If $w + l > u$, then $x = 0$

- Log using RUP rule
- $x = 1$ immediately contradicts $f_r := \Delta + f_o \leq u$

Example

```
rup ~x1
```

Take Home

Done

- Conflict analysis
- LP infeasibility
- Solution-improving optimisation
- Core-guided optimisation

Take Home

Done

- Conflict analysis
- LP infeasibility
- Solution-improving optimisation
- Core-guided optimisation

To Do

- LP integrality cuts
- Complete core-guided optimisation
- Complete deletion

Take Home

Done

- Conflict analysis
- LP infeasibility
- Solution-improving optimisation
- Core-guided optimisation

To Do

- LP integrality cuts
- Complete core-guided optimisation
- Complete deletion

Thanks!