Exploiting Partial-Assignment Enumeration in Optimization Modulo Theories

Gabriele Masina, Roberto Sebastiani

UNIVERSITÀ DI TRENTO
Department of Information Engineering and Computer Science

FroCoS 2025

Optimization Modulo Theories [1]

Given:
- a theory $\mathcal{T}$ defining a total ordering relation $\leq$
- a $\mathcal{T}$-formula φ
- a $\mathcal{T}$-term cost

Goal: find a model $\mathcal{M} \models \varphi$, $\mathcal{M} \models \mathcal{T}$ minimizing cost w.r.t. $\leq$

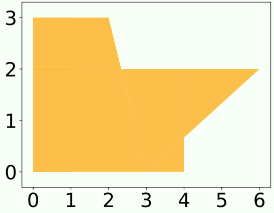
[1] R. Sebastiani and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*

Optimization Modulo Theories [1]

Given:
- a theory $\mathcal{T}$ defining a total ordering relation $\leq$
- a $\mathcal{T}$-formula φ
- a $\mathcal{T}$-term cost

Goal: find a model $\mathcal{M} \models \varphi$, $\mathcal{M} \models \mathcal{T}$ minimizing cost w.r.t. $\leq$

- $$\varphi = \big(0 \leq x \leq 6\big) \wedge \big(0 \leq y \leq 3\big) \wedge \big((2x - 3y \leq 6) \vee (x \leq 4)\big) \wedge \big((y \leq 2) \vee (y \leq -3x + 9) \vee (x < 1)\big)$$



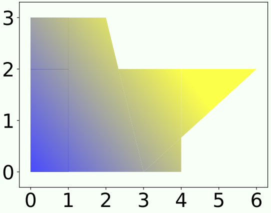
[1] R. Sebastiani and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*

Optimization Modulo Theories [1]

Given: • a theory $\mathcal{T}$ defining a total ordering relation $\leq$
• a $\mathcal{T}$-formula φ
• a $\mathcal{T}$-term cost

Goal: find a model $\mathcal{M} \models \varphi$, $\mathcal{M} \models \mathcal{T}$ minimizing cost w.r.t. $\leq$

- $$\varphi = \big(0 \leq x \leq 6\big) \wedge \big(0 \leq y \leq 3\big) \wedge \big((2x - 3y \leq 6) \vee (x \leq 4)\big) \wedge \big((y \leq 2) \vee (y \leq -3x + 9) \vee (x < 1)\big)$$
- cost $= -(x + y)$



[1] R. Sebastiani and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*

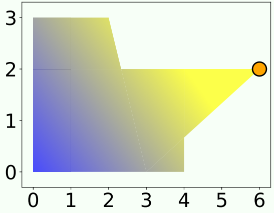
Optimization Modulo Theories [1]

Given:
- a theory $\mathcal{T}$ defining a total ordering relation $\leq$
- a $\mathcal{T}$-formula φ
- a $\mathcal{T}$-term cost

Goal: find a model $\mathcal{M} \models \varphi$, $\mathcal{M} \models \mathcal{T}$ minimizing cost w.r.t. $\leq$

- $$\varphi = \big(0 \leq x \leq 6\big) \wedge \big(0 \leq y \leq 3\big) \wedge \big((2x - 3y \leq 6) \vee (x \leq 4)\big) \wedge \big((y \leq 2) \vee (y \leq -3x + 9) \vee (x < 1)\big)$$

- $\text{cost} = -(x + y)$ $\quad \Longrightarrow \quad$ $\mathcal{M} = \{x \mapsto 6, y \mapsto 2\}$



[1] R. Sebastiani and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*

Which theories?

Lazy approach

$\mathcal{LRA}$	Linear Rational Arithmetic [2]
$\mathcal{LIRA}$	Linear Integer&Rational Arithmetic [3]
$\mathcal{NIRA}$	Nonlinear Arithmetic [4]

[2] R. Sebastiani and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*

[3] R. Sebastiani and P. Trentin (2015). *Pushing the Envelope of Optimization Modulo Theories with Linear-Arithmetic Cost Functions*. In: *TACAS 2015*

[4] F. Bigarella et al. (2021). *Optimization Modulo Non-linear Arithmetic via Incremental Linearization*. In: *FROCOS 2021*

[5] A. Nadel and V. Ryvchin (2016). *Bit-Vector Optimization*. In: *TACAS 2016*

[6] P. Trentin and R. Sebastiani (2021). *Optimization Modulo the Theories of Signed Bit-Vectors and Floating-Point Numbers*. In: *J Autom Reason*

Which theories?

Lazy approach

$\mathcal{LRA}$ Linear Rational Arithmetic [2]
$\mathcal{LIRA}$ Linear Integer&Rational Arithmetic [3]
$\mathcal{NIRA}$ Nonlinear Arithmetic [4]

Eager approach

$\mathcal{BV}$ Bit-Vectors [5]
$\mathcal{FP}$ Floating-Points [6]

---

[2] R. Sebastiani and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*

[3] R. Sebastiani and P. Trentin (2015). *Pushing the Envelope of Optimization Modulo Theories with Linear-Arithmetic Cost Functions*. In: *TACAS 2015*

[4] F. Bigarella et al. (2021). *Optimization Modulo Non-linear Arithmetic via Incremental Linearization*. In: *FROCOS 2021*

[5] A. Nadel and V. Ryvchin (2016). *Bit-Vector Optimization*. In: *TACAS 2016*

[6] P. Trentin and R. Sebastiani (2021). *Optimization Modulo the Theories of Signed Bit-Vectors and Floating-Point Numbers*. In: *J Autom Reason*

Which theories?

Lazy approach

$\mathcal{LRA}$ Linear Rational Arithmetic [2]
$\mathcal{LIRA}$ Linear Integer&Rational Arithmetic [3]
$\mathcal{NIRA}$ Nonlinear Arithmetic [4]

Focus of this talk!

Eager approach

$\mathcal{BV}$ Bit-Vectors [5]
$\mathcal{FP}$ Floating-Points [6]

---

2 R. Sebastiani and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*

3 R. Sebastiani and P. Trentin (2015). *Pushing the Envelope of Optimization Modulo Theories with Linear-Arithmetic Cost Functions*. In: *TACAS 2015*

4 F. Bigarella et al. (2021). *Optimization Modulo Non-linear Arithmetic via Incremental Linearization*. In: *FROCOS 2021*

5 A. Nadel and V. Ryvchin (2016). *Bit-Vector Optimization*. In: *TACAS 2016*

6 P. Trentin and R. Sebastiani (2021). *Optimization Modulo the Theories of Signed Bit-Vectors and Floating-Point Numbers*. In: *J Autom Reason*

Lazy OMT Solving — Basic Schema

$\mathcal{M} \leftarrow \emptyset$ // Best model found so far
ub $\leftarrow \infty$ // Current upper bound
res $\leftarrow$ SAT // Status of the search
$\eta \leftarrow \emptyset$ // T-SAT truth assignment
while res $=$ SAT **do**
  $\langle \text{res}, \eta \rangle \leftarrow$ SMT.IncrementalSolve($\varphi \wedge$ (cost $<$ ub))
  if res $=$ SAT **then**
    $\mathcal{M} \leftarrow$ T-Solver.Minimize(η, cost)
    ub $\leftarrow \mathcal{M}$(cost)
return $\langle$SAT, $\mathcal{M}\rangle$ **if** res $=$ SAT **else** $\langle$UNSAT, $\mathcal{M}\rangle$

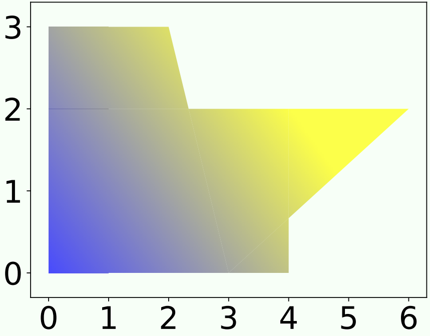
Lazy OMT Solving — Basic Schema

$\mathcal{M} \leftarrow \emptyset$ // Best model found so far
ub $\leftarrow \infty$ // Current upper bound
res $\leftarrow$ SAT // Status of the search
$\eta \leftarrow \emptyset$ // T-SAT truth assignment
while res $=$ SAT **do**
    $\langle$res, $\eta\rangle \leftarrow$ SMT.IncrementalSolve($\varphi \wedge$ (cost $<$ ub))
    if res $=$ SAT **then**
        $\mathcal{M} \leftarrow$ T-Solver.Minimize(η, cost)
        ub $\leftarrow \mathcal{M}$(cost)
return $\langle$SAT, $\mathcal{M}\rangle$ **if** res $=$ SAT **else** $\langle$UNSAT, $\mathcal{M}\rangle$

Can be stopped for anytime optimization!

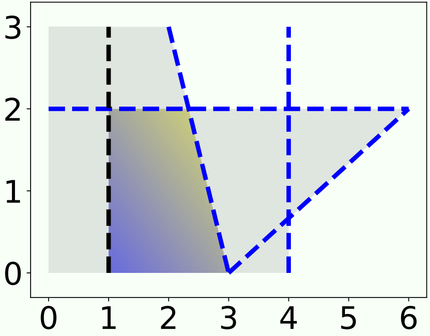
- $\varphi = \big(0 \le x \le 6\big) \land \big(0 \le y \le 3\big)$
  $\land\Big((2x - 3y \le 6) \lor (x \le 4)\Big) \land \Big((y \le 2) \lor (y \le -3x + 9) \lor (x < 1)\Big)$

- $\text{cost} = -(x + y)$



- $\varphi = \big(0 \le x \le 6\big) \wedge \big(0 \le y \le 3\big)$
  $\wedge\Big(\big(2x - 3y \le 6\big) \vee \big(x \le 4\big)\Big) \wedge \Big(\big(y \le 2\big) \vee \big(y \le -3x + 9\big) \vee \big(x < 1\big)\Big)$

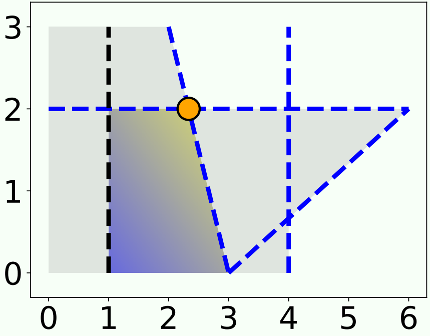
- $\text{cost} = -(x + y)$



$$\eta = \{ \quad (0 \le x \le 6), (0 \le y \le 3), (2x - 3y \le 6), (x \le 4), (y \le 2), (y \le -3x + 9), \neg(x < 1) \quad \}$$

- $\varphi = \big(0 \leq x \leq 6\big) \wedge \big(0 \leq y \leq 3\big)$
  $\wedge\Big((2x - 3y \leq 6) \vee (x \leq 4)\Big) \wedge \Big((y \leq 2) \vee (y \leq -3x + 9) \vee (x < 1)\Big)$

- $\text{cost} = -(x + y)$

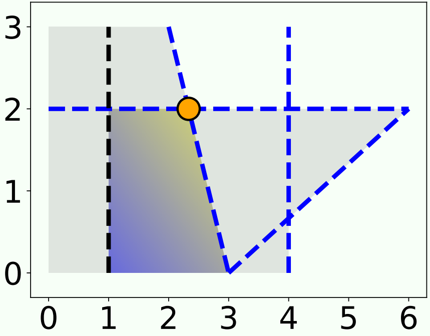


$$\eta = \{ \quad (0 \leq x \leq 6), (0 \leq y \leq 3), (2x - 3y \leq 6), (x \leq 4), (y \leq 2), (y \leq -3x + 9), \neg(x < 1) \quad \}$$

$$\Longrightarrow \mathcal{M} = \{x \mapsto \tfrac{7}{3}, y \mapsto 2\} \text{ (optimum for } \eta)$$

- $\varphi = \big(0 \le x \le 6\big) \wedge \big(0 \le y \le 3\big)$
  $\wedge\Big((2x - 3y \le 6) \vee (x \le 4)\Big) \wedge \Big((y \le 2) \vee (y \le -3x + 9) \vee (x < 1)\Big)$

- $\text{cost} = -(x + y)$



$$\eta = \{ \quad (0 \le x \le 6), \; (0 \le y \le 3), \; (2x - 3y \le 6), \; (x \le 4), \; (y \le 2), \; (y \le -3x + 9), \; \neg(x < 1) \quad \}$$

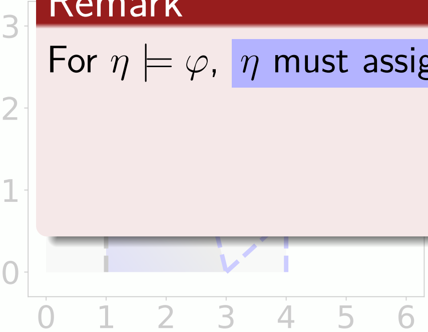
$\Longrightarrow$ $\mathcal{M} = \{x \mapsto \frac{7}{3}, y \mapsto 2\}$ (optimum for η)

$\Longrightarrow$ add $(\text{cost} < -\frac{13}{3})$ and repeat!

- $\varphi = (0 \leq x \leq 6) \wedge (0 \leq y \leq 3)$
  $\wedge\big((2x - 3y \leq 6) \vee (x \leq 4)\big) \wedge \big((y \leq 2) \vee (y \leq -3x + 9) \vee (x < 1)\big)$
- $\text{cost} = -(x + y)$

Remark

For $\eta \models \varphi$, η must assign (at least) one true literal per clause

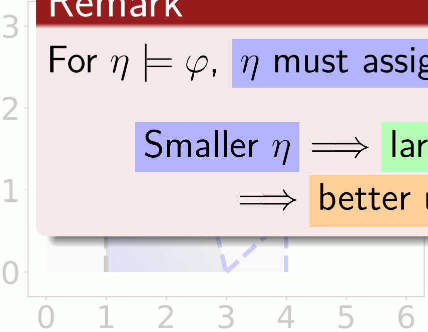


$\Longrightarrow$ $\mathcal{M} = \{x \mapsto \frac{7}{3}, y \mapsto 2\}$ (optimum for η)

$\Longrightarrow$ add $(\text{cost} < -\frac{13}{3})$ and repeat!

- $\varphi = (0 \le x \le 6) \wedge (0 \le y \le 3)$
  $\wedge\big((2x - 3y \le 6) \vee (x \le 4)\big) \wedge \big((y \le 2) \vee (y \le -3x + 9) \vee (x < 1)\big)$

- $\text{cost} = -(x + y)$



Remark

For $\eta \models \varphi$, η must assign (at least) one true literal per clause

Smaller η $\Longrightarrow$ larger solutions space $\Longrightarrow$ better $\mathcal{M}$
$\Longrightarrow$ better ub $\Longrightarrow$ faster convergence!

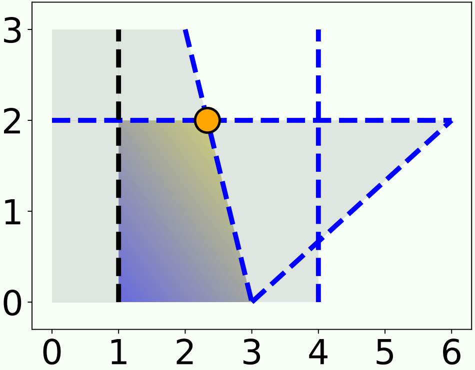
$\Longrightarrow$ $\mathcal{M} = \{x \mapsto \frac{7}{3}, y \mapsto 2\}$ (optimum for η)

$\Longrightarrow$ add $(\text{cost} < -\frac{13}{3})$ and repeat!

Can we exploit partial truth assignments for optimization?

- $\varphi = (0 \le x \le 6) \land (0 \le y \le 3)$
  $\land\big((2x - 3y \le 6) \lor (x \le 4)\big) \land \big((y \le 2) \lor (y \le -3x + 9) \lor (x < 1)\big)$

- $\text{cost} = -(x + y)$



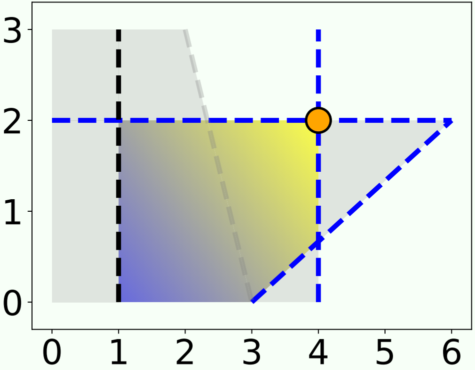
$$\mu = \{ \quad (0 \le x \le 6), \; (0 \le y \le 3), \; (2x - 3y \le 6), \; (x \le 4), \; (y \le 2), \; (y \le -3x + 9), \; \neg(x < 1) \quad \}$$

$\Longrightarrow$ $\mathcal{M} = \{x \mapsto \frac{7}{3}, y \mapsto 2\}$ (optimum for μ)

$\Longrightarrow$ add $(\text{cost} < -\frac{13}{3})$ and repeat!

- $\varphi = (0 \leq x \leq 6) \wedge (0 \leq y \leq 3)$
  $\wedge\big((2x - 3y \leq 6) \vee (x \leq 4)\big) \wedge \big((y \leq 2) \vee (y \leq -3x + 9) \vee (x < 1)\big)$

- $\mathrm{cost} = -(x + y)$



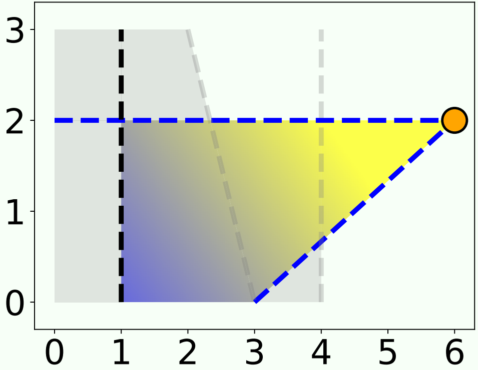
$$\mu = \{ \quad (0 \leq x \leq 6), \; (0 \leq y \leq 3), \; (2x - 3y \leq 6), \; (x \leq 4), \; (y \leq 2), \; (y \leq -3x + 9), \; \neg(x < 1) \quad \}$$

$\Longrightarrow$ $\mathcal{M} = \{x \mapsto 4, y \mapsto 2\}$ (optimum for μ)

$\Longrightarrow$ add $(\mathrm{cost} < -6)$ and repeat!

- $\varphi = (0 \leq x \leq 6) \wedge (0 \leq y \leq 3)$
  $\wedge\big((2x - 3y \leq 6) \vee (x \leq 4)\big) \wedge \big((y \leq 2) \vee (y \leq -3x + 9) \vee (x < 1)\big)$

- $\text{cost} = -(x + y)$



$$\mu = \{ \quad (0 \leq x \leq 6), (0 \leq y \leq 3), (2x - 3y \leq 6), (x \leq 4), (y \leq 2), (y \leq -3x + 9), \neg(x < 1) \quad \}$$

$\Longrightarrow$ $\mathcal{M} = \{x \mapsto 6, y \mapsto 2\}$ (optimum for μ)

$\Longrightarrow$ add $(\text{cost} < -8)$ and repeat!

Lazy OMT Solving with Partial Assignments

$\mathcal{M} \leftarrow \emptyset$ // Best model found so far
ub $\leftarrow \infty$ // Current upper bound
res $\leftarrow$ SAT // Status of the search
$\eta \leftarrow \emptyset$ // T-SAT truth assignment
while res $=$ SAT **do**
  $\langle \text{res}, \eta \rangle \leftarrow$ SMT.IncrementalSolve($\varphi \wedge$ (cost $<$ ub))
  if res $=$ SAT **then**
    $\mu \leftarrow$ OMT-REDUCE-ASSIGNMENT(φ, η, cost)
    $\mathcal{M} \leftarrow$ T-Solver.Minimize(μ, cost)
    ub $\leftarrow \mathcal{M}$(cost)
return $\langle \text{SAT}, \mathcal{M} \rangle$ **if** res $=$ SAT **else** $\langle \text{UNSAT}, \mathcal{M} \rangle$

How to choose the constraints to remove?

Basic Reduction

```
1: μ ← η
2: for ℓ ∈ η do
3:     if μ \ {ℓ} satisfies all clauses in φ then
4:         μ ← μ \ {ℓ}
5: return μ
```

Basic Reduction

```
μ ← η
for ℓ ∈ η do
    if μ \ {ℓ} satisfies all clauses in φ then
        μ ← μ \ {ℓ}
return μ
```

- Works for any OMT theory $\mathcal{T}$!

Basic Reduction

1: $\mu \leftarrow \eta$
2: **for** $\ell \in \eta$ **do**
3: **if** $\mu \setminus \{\ell\}$ satisfies all clauses in φ **then**
4: $\mu \leftarrow \mu \setminus \{\ell\}$
5: **return** μ

- Works for any OMT theory $\mathcal{T}$!
- But may blindly remove "unnecessary" constraints...

Optimization-Aware Reduction

The $\mathcal{T}$-minimizer suggests the constraints "limiting" the optimum
$\implies$ Their removal will very likely lead to a better model!

Optimization-Aware Reduction

The $\mathcal{T}$-minimizer suggests the constraints "limiting" the optimum
$\Longrightarrow$ Their removal will very likely lead to a better model!

$\mu \leftarrow \eta$
$\mathcal{M} \leftarrow$ T-Solver.Minimize(μ, cost)
$\ell \leftarrow$ T-Solver.ProposeLiteralToDrop()
while $\ell \neq \bot$ **do**
  if $\mu \setminus \{\ell\}$ satisfies all clauses in φ **then**
    $\mu \leftarrow \mu \setminus \{\ell\}$
    $\mathcal{M} \leftarrow$ T-Solver.Minimize(μ, cost)
  $\ell \leftarrow$ T-Solver.ProposeLiteralToDrop()
return μ

How to identify these constraints?

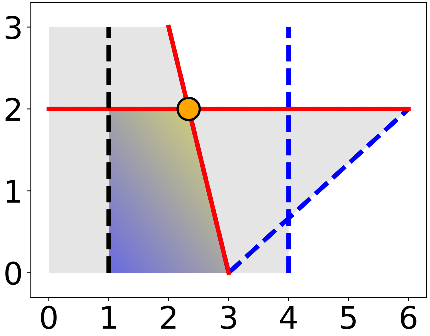
- General way: find (minimal) conflict set for $\mu \wedge (\text{cost} < \mathcal{M}(\text{cost}))$

How to identify these constraints?

- General way: find (minimal) conflict set for $\mu \wedge (\text{cost} < \mathcal{M}(\text{cost}))$
- Ad-hoc way for specific theories: $\mathcal{LRA}$ and $\mathcal{LIRA}$

Optimization-Aware Reduction for OMT($\mathcal{LRA}$)

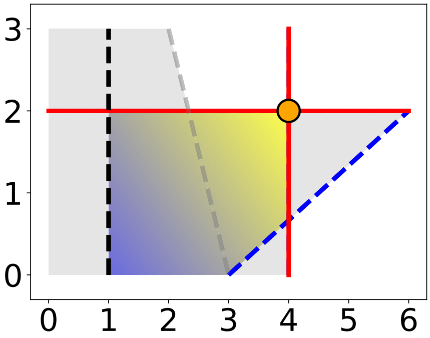
In $\mathcal{LRA}$ the optimum is on a vertex of the polytope defined by μ!
$\implies$ vertex-defining constraints obtained from the Simplex algorithm



$$\mu = \{ \quad \begin{array}{l} (0 \le x \le 6), \\ (0 \le y \le 3), \\ (2x - 3y \le 6), \\ (x \le 4), \\ (y \le 2), \\ (y \le -3x + 9), \\ \neg(x < 1) \quad \} \end{array}$$

Optimization-Aware Reduction for OMT($\mathcal{LRA}$)

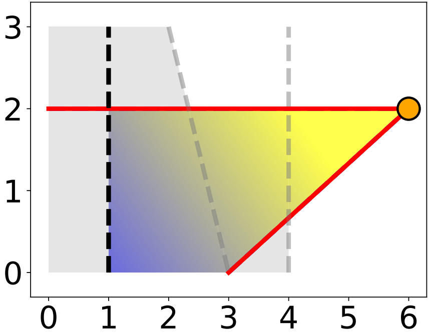
In $\mathcal{LRA}$ the optimum is on a vertex of the polytope defined by μ!
$\implies$ vertex-defining constraints obtained from the Simplex algorithm



$$\mu = \{ \quad \begin{array}{l} (0 \leq x \leq 6), \\ (0 \leq y \leq 3), \\ (2x - 3y \leq 6), \\ (x \leq 4), \\ (y \leq 2), \\ (y \leq -3x + 9), \\ \neg(x < 1) \quad \} \end{array}$$

Optimization-Aware Reduction for OMT($\mathcal{LRA}$)

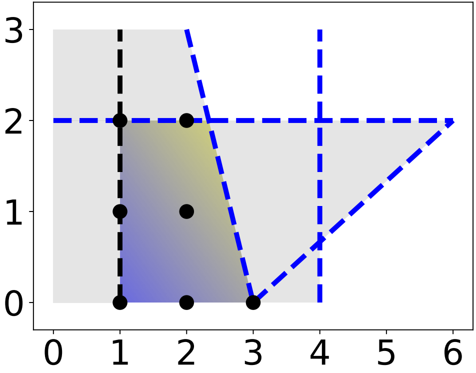
In $\mathcal{LRA}$ the optimum is on a vertex of the polytope defined by μ!
$\Longrightarrow$ vertex-defining constraints obtained from the Simplex algorithm



$$\mu = \{ \quad (0 \leq x \leq 6), \\ (0 \leq y \leq 3), \\ (2x - 3y \leq 6), \\ (x \leq 4), \\ (y \leq 2), \\ (y \leq -3x + 9), \\ \neg(x < 1) \quad \}$$

Optimization-Aware Reduction for OMT($\mathcal{LIRA}$)

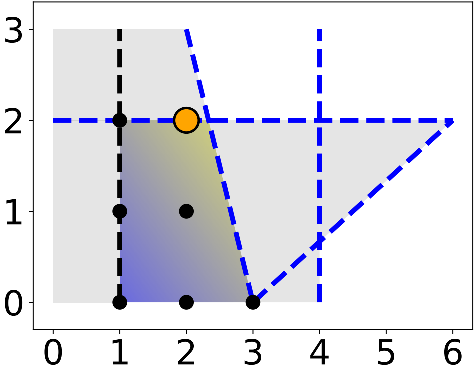
OMT($\mathcal{LIRA}$): reason on the continuous relaxation of the problem!



Optimization-Aware Reduction for OMT($\mathcal{LIRA}$)

OMT($\mathcal{LIRA}$): reason on the continuous relaxation of the problem!

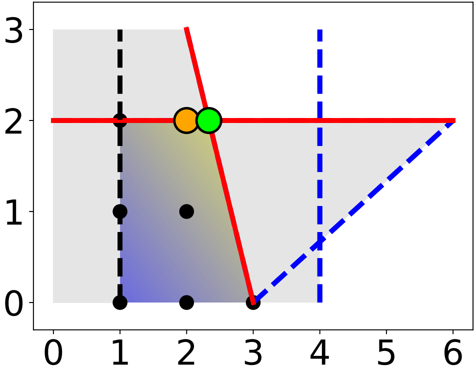
$\Longrightarrow$ the relaxation guides the search for the integer optimum!



Optimization-Aware Reduction for OMT($\mathcal{LIRA}$)

OMT($\mathcal{LIRA}$): reason on the continuous relaxation of the problem!

$\Longrightarrow$ the relaxation guides the search for the integer optimum!



Experimental Evaluation

Experimental Setting

- Implemented in OptiMathSAT [7]
- Benchmarks
  - OMT($\mathcal{LRA}$): Strip-Packing [8], Temporal Planning [9]
  - OMT($\mathcal{LIRA}$): Strip-Packing
  - OMT($\mathcal{LRA} \cup \mathcal{AR}$): Strip-Packing
- Compare
  - Running time (time (s))
  - Quality of the best model found within the time limit of 1200s (upper bound)

[7] R. Sebastiani and P. Trentin (2020). *OptiMathSAT: A Tool for Optimization Modulo Theories*. In: *J Autom Reason*

[8] R. Sebastiani and S. Tomasi (2015). *Optimization Modulo Theories with Linear Rational Costs*. In: *ACM Trans. Comput. Logic*

[9] S. Panjkovic and A. Micheli (2024). *Abstract Action Scheduling for Optimal Temporal Planning via OMT*. In: *AAAI 2024*

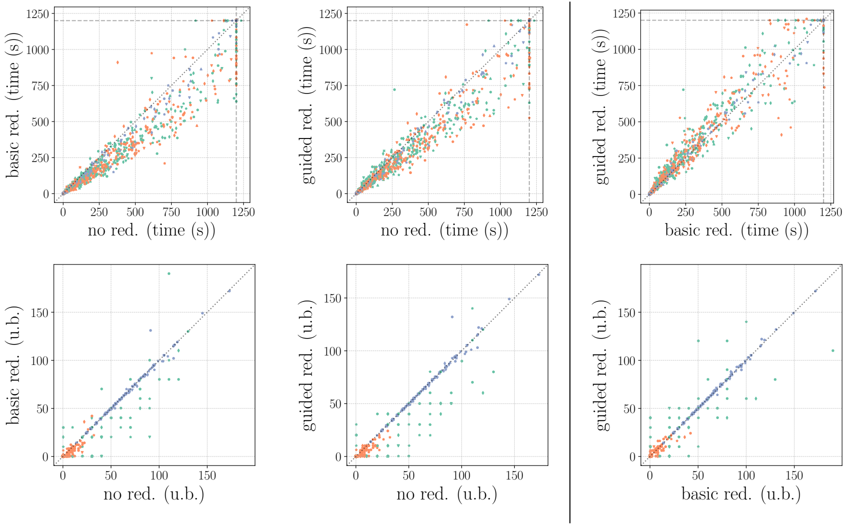


Figure 1: Results for OMT($\mathcal{LRA}$) Temporal Planning (1520 problems). Timeouts: 246 no red., 211 basic red., 212 guided red.

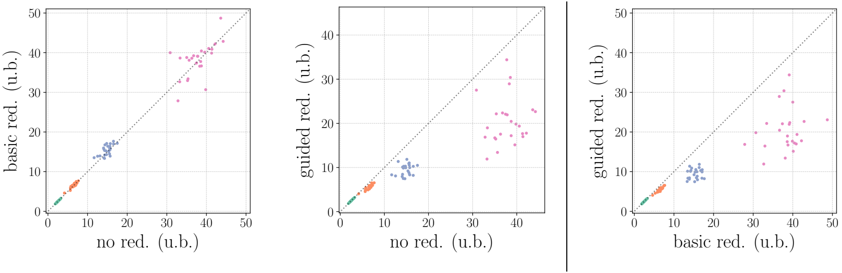


Figure 2: Results for OMT($\mathcal{LRA}$) Strip-Packing (100 problems, all timed out).

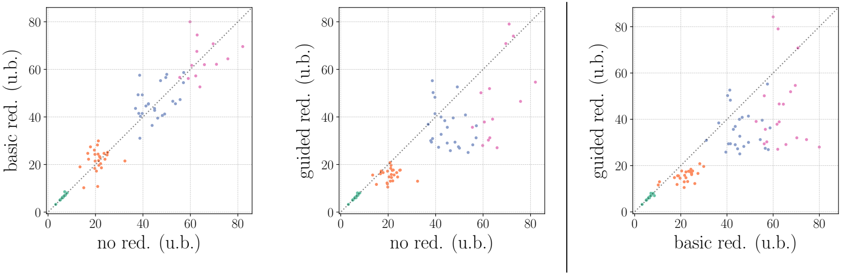


Figure 3: Results for OMT($\mathcal{LIRA}$) Strip-Packing (100 problems, all timed out).

Conclusions & Takeaways

- Exploiting partial assignments in OMT is a promising direction for both optimal and anytime OMT solving!
- Optimization-aware reductions are very effective in practice!

Future work

- Ad-hoc optimization-aware procedures for other theories
- More advanced reduction techniques (e.g., entailment-based [10])

[10] R. Sebastiani (2025-30). *Entailment vs. Verification for Partial-Assignment Satisfiability and Enumeration*. In: *CADE 30*

Questions?

Bibliography I

- Bigarella, F. et al. (2021). *Optimization Modulo Non-linear Arithmetic via Incremental Linearization*. In: *FROCOS 2021*. LNCS. Springer, pp. 213–231.
- Nadel, A. and V. Ryvchin (2016). *Bit-Vector Optimization*. In: *TACAS 2016*. LNCS. Springer, pp. 851–867.
- Panjkovic, S. and A. Micheli (2024). *Abstract Action Scheduling for Optimal Temporal Planning via OMT*. In: *AAAI 2024* 38.18, pp. 20222–20229.

Bibliography II

- Sebastiani, R. (2025-07-30). *Entailment vs. Verification for Partial-Assignment Satisfiability and Enumeration*. In: *CADE 30*. Berlin, Heidelberg: Springer-Verlag, pp. 717–735.
- Sebastiani, R. and S. Tomasi (2012). *Optimization in SMT with LA(Q) Cost Functions*. In: *IJCAR 2012*. Vol. 7364. LNCS. Springer, pp. 484–498.
- Sebastiani, R. and S. Tomasi (2015). *Optimization Modulo Theories with Linear Rational Costs*. In: *ACM Trans. Comput. Logic* 16.2, 12:1–12:43.

Bibliography III

- Sebastiani, R. and P. Trentin (2015). *Pushing the Envelope of Optimization Modulo Theories with Linear-Arithmetic Cost Functions*. In: *TACAS 2015*. LNCS. Springer, pp. 335–349.
- Sebastiani, R. and P. Trentin (2020). *OptiMathSAT: A Tool for Optimization Modulo Theories*. In: *J Autom Reason* 64.3, pp. 423–460.
- Trentin, P. and R. Sebastiani (2021). *Optimization Modulo the Theories of Signed Bit-Vectors and Floating-Point Numbers*. In: *J Autom Reason* 65.7, pp. 1071–1096.