# CEGAR-Based Approach for Solving Combinatorial Optimization Modulo Quantified Linear Arithmetics Problems

cnrs · IRISA · Inria · LaBRI · Université de Rennes · université de BORDEAUX

## Optimization Modulo Quantified Linear Arithmetics Problems — OPT+qLP

minimize $f_{\text{obj}}(x)$ | minimizing a Boolean objective function $f_{\text{obj}} : \mathbb{B}^n \to \mathbb{R}$

such that:

$$\bigwedge_c c(x)$$

**SAT problem**: SAT / ASP solvers, Boolean Optimization

$$\wedge \bigwedge_d d(x,y)$$

**SAT problem modulo linear constraints**: SMT solvers (*QF_LRA*), *Clingo[lpx]* **[1]**

$$\wedge\ \forall z \in \mathbb{R}^p, \bigwedge_e e(x,z) \implies \bigwedge_h h(x,z)$$

**SAT problem modulo quantified linear constraints**: *Restricted to one level of quantifiers*

SMT solvers and *Clingo[lpx]* with quantifier elimination

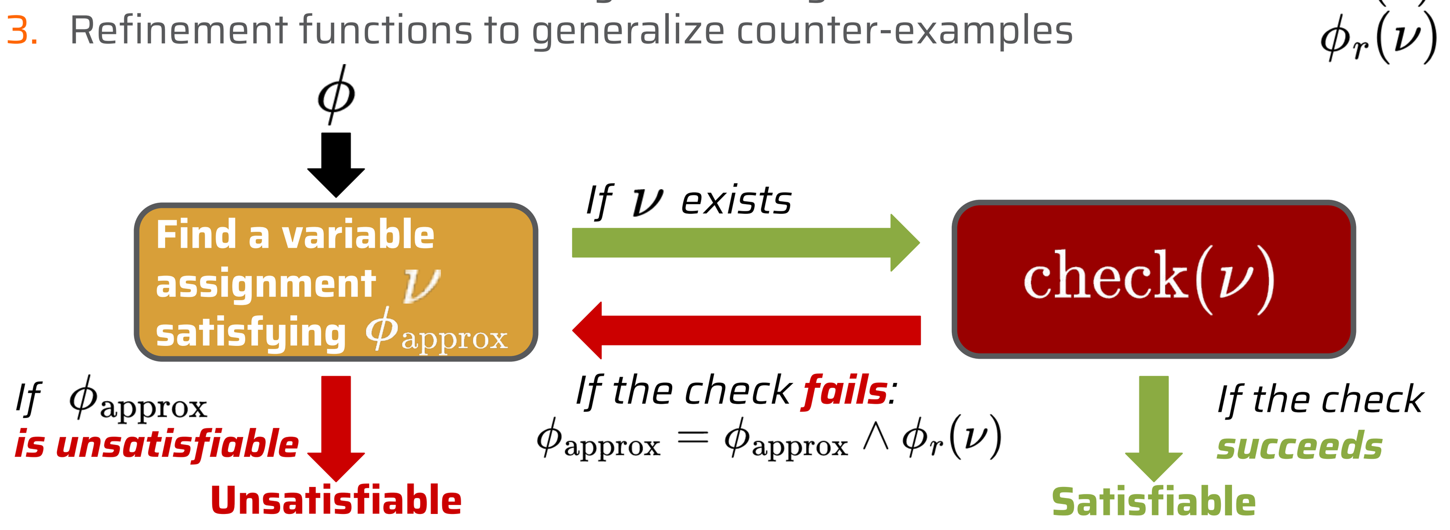
with $x \in \mathbb{B}^n, y \in \mathbb{R}^m$ $c : \bigvee_i x_i \vee \bigvee_j \neg x_j$

$d, e, h : \bigvee_i x_i \vee \bigvee_j \neg x_j \vee \sum_k \lambda_k \times y_k \leq 0$

## CEGAR [2]: *Counter-Example Guided Abstraction Refinement*

**Rely on:**

1. An over-approximation of the OPT+qLP problem — $\phi \implies \phi_{\text{approx}}$
2. Methods to check the validity of an assignment — $\text{check}(\nu)$
3. Refinement functions to generalize counter-examples — $\phi_r(\nu)$




- Similar to offline SMT approaches
- Already used to solve SMT problems **[3]**

**Advantages:**

- Solver independent
- Efficient for combinatorial problems

## Contribution: A CEGAR for Solving OPT+qLP problems

### 1. Over-Approximation

- Replace each linear constraint by an unique Boolean variable
- Replace $\implies$ by a $\wedge$ in the universally quantified linear constraint

### 2. Checking quantified linear constraints

**Definition**

$\mathcal{C}_i$: set of LP constraints that must hold for an assignment $x$ of $\phi_{\text{approx}}$ to satisfy $\bigwedge_i i(x,y)$

**Existential quantifier**

If $\mathcal{C}_d$ is satisfiable **accept**, else **reject**

**Universal quantifier**

1. Check if $\mathcal{C}_e$ is satisfiable
   a. if **no**, **accept**
   b. if **yes**, continue
2. For each $h \in \mathcal{C}_h$:
   a. $h^*$: maximum of $h$ under $\mathcal{C}_e$
   b. if **not** $h^* \leq 0$, **reject**
3. **accept**

***Note:*** *Linear optimization problems are solved by dedicated LP solvers*

### 3. Counter-example generalization

**Rely on 2 monotone properties:**

1. All supersets of an unsatisfiable set of linear constraints is unsatisfiable

**Unsatisfiable core:** smallest unsatisfiable subset of linear constraints

2. Adding constraints to a linear optimization problem cannot increase its maximum value

**Optimal core of** $(h, \mathcal{C})$: biggest superset of $\mathcal{C}$ having the same maximum value as $(h, \mathcal{C})$

**Example:** $\{\alpha,\beta\}$ and $\{\alpha,\gamma\}$ are optimal cores of $\{\alpha\}$
All their subsets are such that $\max y = \infty$

**Problem:** *computing **optimal** and **unsatisfiable cores** can be computationally expensive*

**Proposition: *linear constraint partitioning***
Partition the set of linear constraints into independent subsets, *i.e. no shared variables between subsets*

**EXAMPLE**

**OPT+qLP problem** $\phi$

$(a \vee b \vee c)$

$\wedge\ \forall x, y \in \mathbb{R}, \left( \begin{array}{l} (y \geq 1 \vee \neg a) \\ \wedge\ (x + y \leq 1 \vee \neg b) \\ \wedge\ (-x + y \leq 0 \vee \neg c) \end{array} \right) \implies y \leq 0.6$

with $a, b, c \in \mathbb{B}$

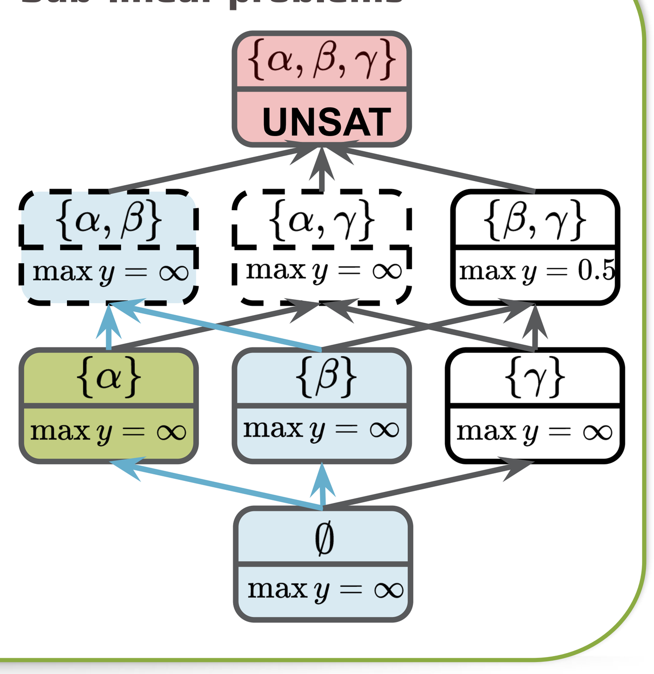
**Its over-approximation** $\phi_{\text{approx}}$

$(a \vee b \vee c)$

$\wedge\ (\alpha \vee \neg x_1) \wedge (\beta \vee \neg x_2) \wedge (\gamma \vee \neg x_3) \wedge \delta$

with $\alpha, \beta, \gamma, \delta \in \mathbb{B}$

**Sub-linear problems**




## Implementation and Benchmark

### Our implementation: *MerrinASP*

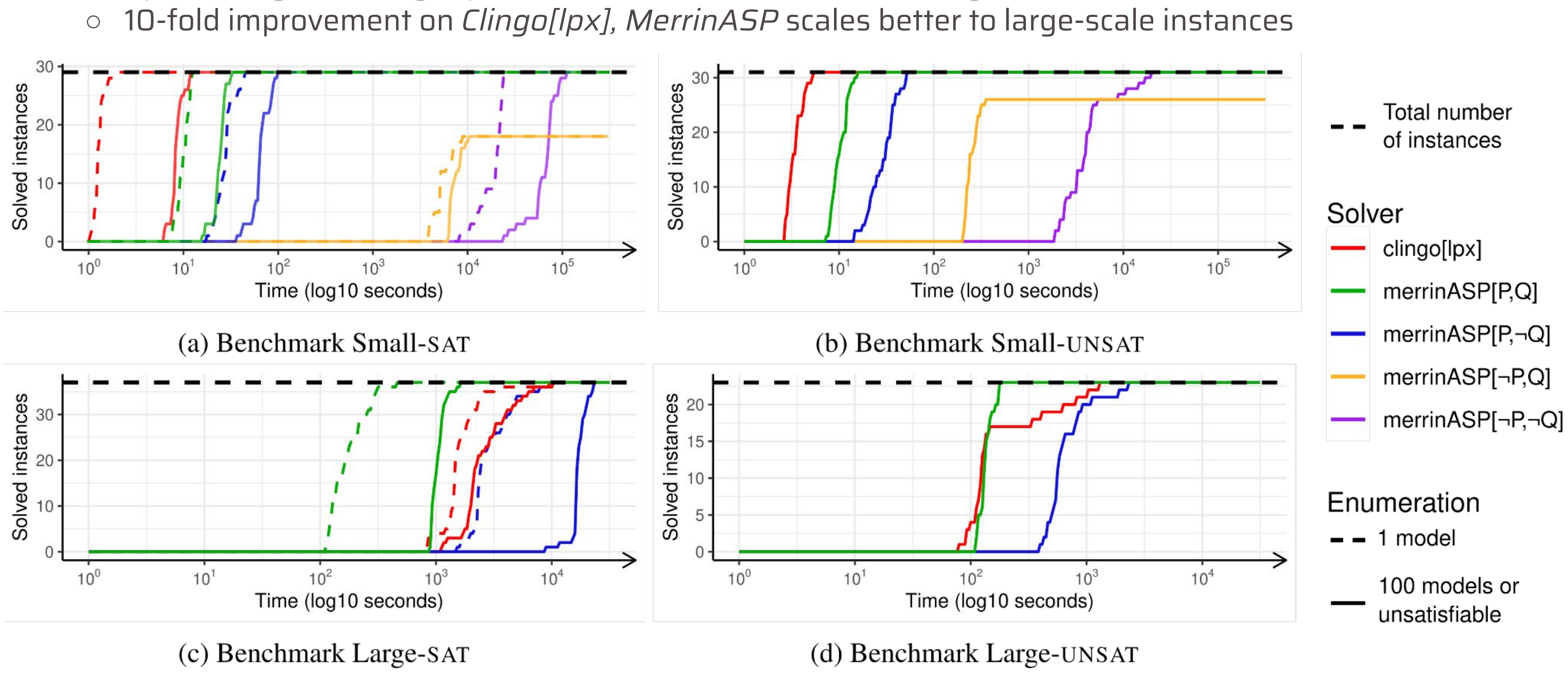
Extend ASP solver *clingo* **[4]** with:

- one-level of quantified linear constraints ;
- linear constraints partitioning.

Support different LP solvers (*e.g.* CPLEX, GUROBI, GLPK, etc).

### Benchmarks

- 2 benchmarks, each composed of 60 OPT+qLP problems inspired by System Biology **[5]**
- Comparison against *Clingo[lpx]* **[1]** and *MerrinASP* under 4 configurations
  - 10-fold improvement on *Clingo[lpx], MerrinASP* scales better to large-scale instances




(a) Benchmark Small-SAT (b) Benchmark Small-UNSAT (c) Benchmark Large-SAT (d) Benchmark Large-UNSAT

## Conclusion

- CEGAR-based approach to solve OPT+qLP problems
- Validate on a benchmark inspired by Bioinformatics
- *MerrinASP*: implementation available on *GitHub*

### Future Work

- Study the impact of line ar solver and its interface on performance
- Compare generate constraints with constraints generated by the LRA theory


Kerian THUILLIER
Univ. Rennes, Inria, CNRS, IRISA, Rennes, France

Anne SIEGEL
Univ. Rennes, Inria, CNRS, IRISA, Rennes, France

Loïc PAULEVÉ
Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, Talence, France

**Contacts**
kerian.thuillier@irisa.fr
anne.siegel@irisa.fr
loic.pauleve@labri.fr


***MerrinASP repository*** ***Proceeding***

**References**


**[1]** T. Janhunen et al. TPLP. 2017.
**[2]** E. Clarke et al.. Journal of the ACM. 2003.
**[3]** R. Brummayer et al. International Workshop on Satisfiability Modulo Theories. 2008.
**[4]** M. Gebser et al. CoRR. 2017.
**[5]** K. Thuillier et al. Oxford Bioinformatics. 2022.