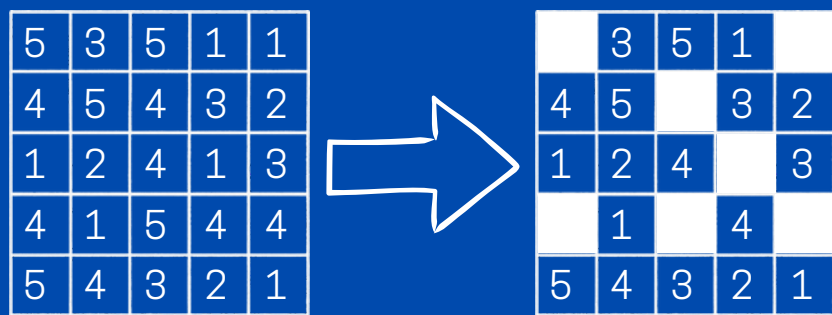


Modelling and solving Hitori using Satisfiability Modulo Theories

1. Background

- **Hitori** is an NP-complete logic puzzle with rules enforcing uniqueness, adjacency and connectivity of tiles
- **Constraint modelling** is the process of translating rules into logical constraints that a solver can reason about
- **Satisfiability Modulo Theories (SMT)** is a solving paradigm extending **SAT** with richer theories, allowing more direct encodings
- **Z3** is a widely used solver that implements the SMT paradigm



2. Objectives

We evaluate the SMT paradigm by modelling and solving Hitori puzzles using Z3 and evaluating:

- The influence of **puzzle size** and **encoding size** on solver performance
- The effects of various **redundant constraints** on solver performance
- The impact of the differing **puzzle structure** of puzzle instances on solver behaviour

3. Approach

Evaluate **7** encodings by **runtime** and **encoding size**

- **3** theory-based encodings
 - Linear Integer Arithmetic (**qf_ia**)
 - Bitvectors (**qf_bv**)
 - Boolean (**qf_bool**)
- **3** alternate variants of **qf_ia**
- **1** alternate approach with connectivity checked outside the solver (**qf_ia_external**)

Evaluate **16** redundant constraints, individually added to **qf_ia**

- Human solving strategies and patterns
- Constraints to enforce connectivity

Evaluate 2 methods to identify **difficult** puzzles

- Runtime-based
- Effort-based (conflicts, decisions, propagations)

4. Experimental results

4.1 Solver runtime and encoding size scaling with puzzle size

Evaluated set: 150 puzzles for each puzzle size $n \in \{5, \dots, 25\}$

- Figure 1 shows runtime by puzzle size (a) and encoding size by puzzle size (b) for all evaluated encodings
- The **qf_ia** encoding achieves the best and most stable runtime across all puzzle sizes
 - Encodings using alternative theories scale much worse than **qf_ia** and its variants
 - Alternate **qf_ia**-based encodings show slightly worse scaling than the base **qf_ia** encoding
- Solver runtime shows strong correlation with encoding size for all encodings
- Additional influences on runtime are most noticeable in:
 - **qf_bv**: likely due to additional overhead from the bitvector theory
 - **qf_ia_external**: solution guessing leads to worse scaling as the search space grows

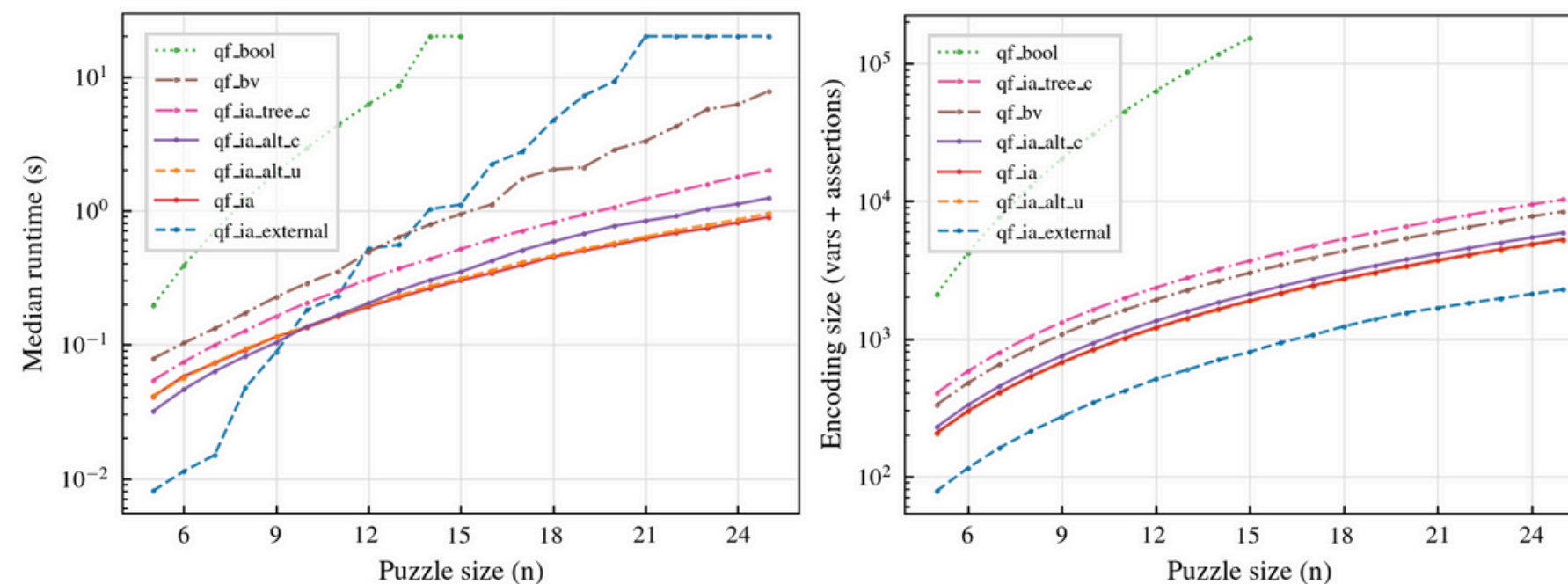


Fig. 1. Evaluation of solver runtime and encoding size for 7 SMT encodings on a logarithmic y-axis

4.2 Impact of redundant constraints on solver performance

Evaluated set: 150 puzzles for each puzzle size $n \in \{5, \dots, 25\}$

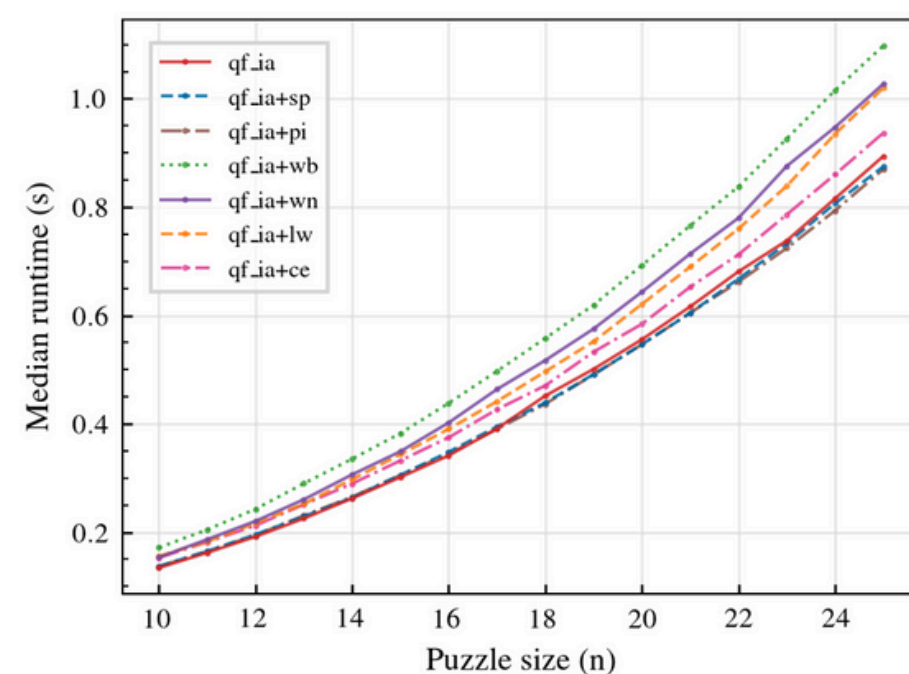


Fig. 2. Median runtime per puzzle size for a selection of redundant constraints added to the **qf_ia** encoding

When added to the **qf_ia** encoding:

- Most redundant constraints slow down solver performance
 - Only 2 constraints show statistically significant speed-ups for $n > 17$
 - Magnitudes of these speed-ups are small, reaching only 2-3% at $n = 25$
- The worst performing constraints show the largest increase in encoding size

When added to the **qf_ia_external** encoding:

- Some redundant constraints improve performance
- Improvements occur because redundant constraints compensate for missing information

4.3 Influence of puzzle structure on solver performance

Evaluated set: 5000 puzzles of size $n = 8$

- Runtime-based filtering shows no correlation between structural properties and solver runtime
- Effort-based filtering reveals some statistically significant differences in structural properties
 - Correlations for these properties are only weak to moderate
 - Other properties show no correlation
- Overall, structural puzzle properties have little impact in solver performance compared to puzzle size and encoding size

Feature	Mean occurrences			Correlation coefficient	
	Baseline	Effort	Runtime	Effort	Runtime
Black tiles	18.2	17.7	18.3	-0.328	0.001
Adjacent pair duplicates	8.3	8.1	8.4	-0.007	0.018
Adjacent triple duplicates	0.2	0.3	0.4	-0.003	0.016
Non-adjacent duplicates	19.2	18.2	19.1	-0.126	0.005
Row/column duplicates	15	14.2	14.8	-0.093	0.027

Table 1. Structural properties for baseline, effort-hard and runtime-hard puzzles. with Spearman's correlation coefficients

5. Conclusion

- SMT is a viable method for solving Hitori, **consistently solving** all evaluated puzzle sizes
- Hitori rules can be naturally expressed using **Linear Integer Arithmetic**, which shows the **best overall performance**
- Solver runtime is mostly impacted by **puzzle size** and **encoding size**
 - Puzzle structure shows **little impact** on solver performance
- **Compact** and **complete** encodings are most effective, without requiring additional modelling complexity or redundant constraints

6. Limitations & Future Work

- Limitations that could motivate future work include:
 - Z3 and theory-specific configuration options were not explored and may influence solver performance
 - Other SMT solvers were not evaluated and may show different behaviour when used with the same encodings
- This study can be used as part of a larger comparison of modelling-and-solving paradigms