

TACTIX: Towards a Programmable Strategy Interface for Flexible SAT Solving

Haoze Wu¹, André Schidler³, Armin Biere³, and Nina Narodytska²

¹ Amherst College, Amherst, MA, USA

² Broadcom, Palo Alto, CA, USA

³ University of Freiburg, Freiburg im Breisgau, Germany

Abstract. Modern SAT solvers combine a rich repertoire of proof rules with sophisticated strategies that determine when and how to apply them. Often, the internal implementations are tightly coupled to particular strategies, making it inconvenient to define new strategies without changing solver internals. We present TACTIX, a proof-producing sequential CDCL solver that incorporates a wide range of techniques in state-of-the-art solvers with a clean separation of proof rules from solving strategies. This separation enables a public interface through which users can invoke particular techniques, observe their effects, and program a wide range of solving strategies. The default strategy of TACTIX is programmable over this interface and outperforms state-of-the-art solvers on benchmarks from the 2024–2026 SAT Competitions. We also use the interface to develop novel strategies tailored to challenging benchmarks from two application domains, which led to performance gains over strong baselines. We intend TACTIX to serve as an infrastructure for designing and evaluating novel solving strategies. Finally, we also present a self-study of the AI-assisted development process behind Tactix.

Keywords: SAT Solving · Solving Strategy · AI-assisted Solver Development

Code: <https://github.com/ai-ar-research/Tactix/>

1 Introduction

Modern SAT solvers achieve impressive practical performance by carefully orchestrating a rich set of techniques, including conflict-driven clause learning, restarts, local search, and various inprocessing rules such as bounded variable elimination [25], clause vivification [54], and bounded variable addition [45]. As the repertoire of reasoning techniques expands, the number of possible ways to combine and apply them grows combinatorially, making it increasingly challenging to identify effective solving strategies. Modern SAT solvers typically employ hard-coded strategies that are carefully tuned on a broad range of benchmarks, with their internal implementations tightly coupled to these strategies to maximize performance. Although solvers often expose configuration options for adjusting aspects of their strategies (e.g., whether to enable particular inprocessing rules or how much computational effort to allocate to a technique), these options still cover only a limited subset of the possible strategies and their combinations, potentially excluding those better suited to specific problem domains of interest. Consequently, in order to experiment with a new solving strategy that significantly deviates from the default,

the user often has to introduce changes to the solver’s internal code and understand the solver’s internal implementation well enough to ensure that their modification does not introduce unintended correctness or performance issues.

Ideally, a SAT solver should cleanly separate the implementation of solving techniques from the strategies that orchestrate their application. Such a separation would allow users to program and experiment with a rich range of strategies without modifying the solver’s internals, potentially making more effective use of the available techniques. It would also facilitate a systematic study of solver behavior and provide a unified framework in which different SAT solvers can be expressed as distinct strategies over a common set of techniques. The separation of proof rules from solving strategies is well established in abstract calculi [51], and its practical appeal has also long been recognized [28, 46]. However, realizing this vision in a practical, high-performance SAT solver raises several non-trivial design questions: at what granularity should the interface expose individual techniques? For example, should users invoke bounded variable elimination on one variable at a time, or should they be able to perform a batch of eliminations governed by a low-level variable selection heuristic and a computational budget? How can we strike a balance between maintaining the efficiency of the implementation and allowing the application of a rich set of techniques at arbitrary stages of the solving process? How can diverse techniques from different solvers be incorporated into a uniform backend?

In this paper, we take a step towards such rule/strategy separation in SAT-solving and introduce TACTIX, a proof-producing sequential CDCL solver with a public interface for defining solving strategies. This interface allows a user to implement a strategy as a client program that selects among the currently available operations, specifies how they should be applied, observes their effects, and decides which operation to invoke next. For example, the strategy may ask to perform bounded variable elimination with specific limits on clause growth and the number of resolution attempts. The TACTIX backend would perform the operation, and the strategy can then inspect why the operation stopped (e.g., completed, budget reached, or encountered a conflict) and the effect of the operation (e.g., how many resolution attempts were performed and how many variables were eliminated). Based on this information, the strategy may repeat the operation or select another operation. Throughout this process, the solver core maintains the formula and search state, emits any proof steps required by a transformation, and records the information needed to reconstruct a satisfying assignment to the original formula. TACTIX incorporates a wide range of search, simplification, symmetry-breaking, and structural reasoning techniques from existing tools, including KISSAT [16], CADICAL [12], SATSUMA [1], CRYPTOMINISAT [64], and LYMPHOSAT [29], and, depending which techniques was invoked, generates UNSAT proofs in DRAT [68], DSR [23], or PBP [18] formats.

TACTIX also comes with a default strategy that is programmable over the public interface. This strategy employs the interval/effort-based rule-scheduling employed in KISSAT. On benchmarks from the 2024–2026 SAT Competitions, this default strategy performs competitively against winners of the 2024–2026 SAT Competitions, solving more instances overall and contributing unique solutions. To further understand the usefulness of the public interface, we co-design with an AI agent new solving strategies over the public interface for two sets of challenging benchmarks, one for arithmetic circuit verification [38, 57], the other for tackling finite cases of the Chvátal’s conjecture [21, 41].

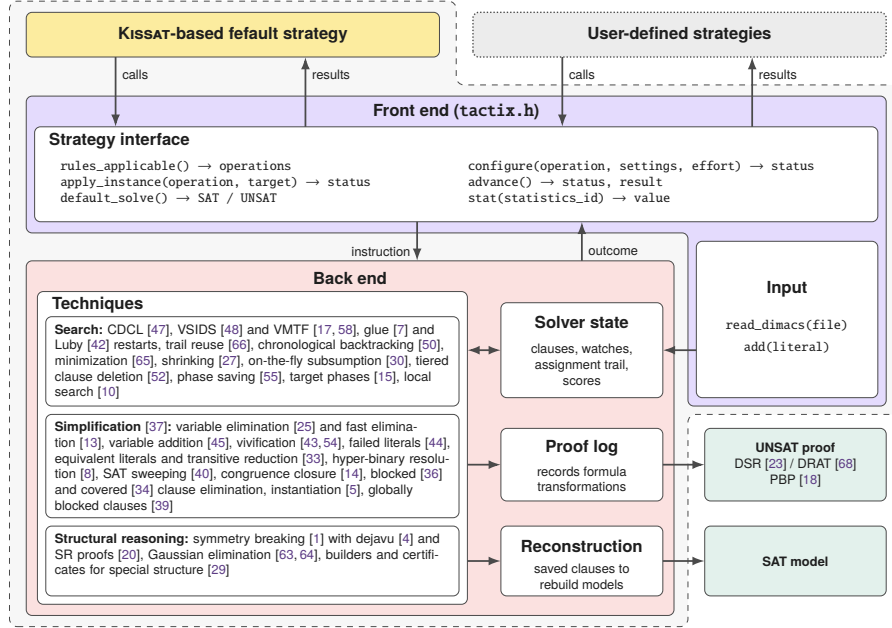


Fig. 1. Tactix system overview with selected interface functions.

In both cases, we were able to come up with a novel strategy that deviates significantly from canonical strategies and achieves substantial gains over baseline solvers in terms of solving and proof-checking time.

In addition to the system description of TACTIX, we also describe our solver development process, including how we perform correctness and performance testing, define milestones, and leverage AI agents. We reflect on the experience of developing TACTIX and discuss infrastructures that are currently under-developed in the formal methods community that can facilitate co-developing solvers with AI agents in a more efficient and systematic manner.

The rest of the paper is organized as follows. Section 2 represents the bulk of the contributions and describes the design of TACTIX, including the backend and the public interface for defining solving strategies. Section 3 describes our AI-assisted software development process. Section 4 presents an empirical evaluation of TACTIX on SAT competition benchmarks, a set of arithmetic verification benchmarks, and subproblems of Chvátal’s conjecture. Section 6 reviews related work. We conclude and discuss future directions in Section 7.

2 TACTIX: A SAT Solver with a Programmable Strategy Interface

This section introduces the overall architecture of TACTIX, describes the operations supported by the backend, and explains how the public interface allows users to compose them into strategies.

2.1 System overview

Figure 1 shows a system overview of TACTIX. TACTIX can be used as an off-the-shelf SAT solver or as a library where a user can create their own solving strategy. The executable reads a formula in DIMACS format and solves it with a default strategy. A library client creates a solver instance, supplies the formula through the input functions of the C interface (reading a DIMACS file or adding clauses), and can either invoke the default strategy or define their own strategies. The dashed boundary encloses the shipped codebase.

The frontend, `tactix.h`, is the public interface. It consists of the input and strategy interfaces, which provide the building blocks for programming solving strategies, including methods for invoking operations, observing their effects (e.g., number of forced literals), and inspecting solver states. The default strategy of TACTIX, which is based on that of KISSAT, can be programmed directly through this public interface.⁴ Each operation can specify its own settings and computational budget (a.k.a. effort). The core applies the selected technique, maintains the solver state, and records proof steps and model-reconstruction data as required. The public interface can also inspect the outcome of the application, which can help decide the next move. The backend implements the operations and maintains the formula, search state, proof, and model-reconstruction data. To support different operation sequences, it checks the conditions required by each technique: for example, unit propagation must converge before a new branching decision is made, and bounded variable elimination can only be applied at decision level zero. An accepted request executes the selected operation and reports the outcome and effects.

The solver produces either SAT or UNSAT. A SAT result comes with a satisfying assignment to the original input variables. When proof production is enabled, an UNSAT result comes with a certificate of unsatisfiability that can be checked externally by a proof checker. The main proof format is deletion substitution redundancy (DSR), which is a superset of the DRAT format [68] and supports ordinary clausal deductions together with satisfiability-preserving restrictions such as symmetry breaking [20, 23]. TACTIX can also sometimes produce pseudo-Boolean proof [18], as discussed later in this section.

2.2 Solver backend

TACTIX's back end integrates CDCL search, simplification, and structural reasoning. Simplification transforms the formula before and between periods of search [25, 37]. Structural reasoning exploits relationships such as symmetries [1] and systems of equations [64] represented by the clauses to derive new clauses or simplify the formula. When the back end receives a request to perform an operation, it checks the applicability of the operation, updates the shared solver state, performs the bookkeeping needed to keep the data structures consistent for subsequent operations, and records the information needed to certify its result. This organization allows a strategy to combine techniques

⁴ The same strategy is also implemented by calling the same operations in the C++ backend, which is used by the compiled TACTIX executable. Empirically (Section 4), the two implementations yield identical behaviors, and the API version only introduces minor performance overhead.

while retaining the progress made by earlier operations. The full list of registered operations is available in the source distribution.⁵

Search and shared state. TACTIX implements a standard CDCL search procedure [47]. It stores the current clauses in one clause database and the sequence of assignments in a *trail*. Each propagated assignment has a *reason*: the clause that forced it. Conflict analysis follows these reasons to derive a learned clause. Assignments are grouped by decision level, which counts the branching decisions currently in effect; level zero contains assignments derived without branching. These structures are shared by all techniques. For example, a clause derived by algebraic reasoning [64] can force an assignment by unit propagation and subsequently serve as a reason during conflict analysis. TACTIX incorporates a number of modern features of CDCL search. These include the VSIDS [48] and variable-move-to-front (VMTF) [17] heuristics for branching, GLUCOSE-style and Luby-style restarts [15, 42], tiered learned-clause deletion [52], and different phase saving techniques [15, 55]. TACTIX also contains a local search module that can be invoked with its own effort limit and search heuristics. While the front end allows the users to define search behavior at fine granularity (e.g., deciding on specific variables rather than using the pre-defined branching heuristics), these pre-defined mechanisms are available as useful building blocks for defining solving strategies.

Simplification. Tactix supports a number of formula simplification techniques from KISSAT and CADICAL, such as subsumption [12, 16] and vivification [54] for clause reduction and simplification, failed-literal probing [44] for look-ahead style variable fixing, congruence closure [14] and SAT sweeping [40] for discovering equivalences and forced literals, substitution of equivalent literals [33], bounded variable elimination [25], and bounded variable addition that introduces fresh variables to factor repeated clause structure and reduce the number of clauses [45]. Similar to how search is implemented, TACTIX incorporates pre-defined low-level heuristics for each technique. For variable elimination, for example, it prioritizes variables using estimates of clause growth and revisits its choices as the formula changes, following KISSAT’s elimination heuristics [13]. While a user could, for example, select individual variables to eliminate or clauses to vivify, it is often reasonable to leave these low-level choices to pre-defined heuristics.

Structural reasoning. In addition to the aforementioned simplification techniques, TACTIX also incorporates techniques that discover specific structures in the formula to derive consequences or guide the search for a model. Specifically, TACTIX incorporates a number of symmetry-breaking techniques including orbitopal fixing, clausal fixing, and negation fixing [1, 2]. It also incorporates Schreier–Sims table cuts, which add binary clauses ordering symmetric literals [3]. Symmetry detection relies on the dejavu graph-automorphism library [4], applied to a graph representation of the current formula. Instead of directly invoking a mature tool like SATSUMA, we implement these techniques within TACTIX’s back end to support strategies that apply symmetry breaking between CDCL searches. In addition, TACTIX detects systems of equations over GF(2) and

⁵ `src/rules/registry.cpp`

applies the Gaussian-elimination approach used in CRYPTO_{MINI}SAT [63, 64], translating deductions back into clauses that search can use. TACTIX also incorporates selected techniques from LYMPHO_{SAT} [29], a SATZILLA-style [70] solver that selects among over 120 specialized solvers. We prioritize techniques that focus on common algebraic and combinatorial structures, such as parity equations, exact-cover constraints, and independent sets, and generalize methods that rely on hard-coded hints. Concretely, we incorporated twelve model-construction-only procedures as local search and seven proof-producing refutation procedures for structures such as parity, cardinality, and Ramsey constraints. Each procedure recognizes constraints encoded by the clauses and operates on the recovered structure, translating its result back into a Boolean variable assignment or a refutation proof. All these additions can be interleaved with search and simplification as they operate on the same backend state.

Proof production. For UNSAT instances, TACTIX by default produces DRAT proofs [68], including for GF(2) reasoning [53]. These are extended to DSR (a superset of DRAT) when techniques such as symmetry breaking require substitution-redundancy steps [20, 23]. The backend checks each substitution-redundancy step before applying it, either against the current formula or leveraging previously verified structural properties such as symmetries. TACTIX contains two preprocessing procedures derived from LYMPHO_{SAT} that produce pseudo-Boolean (PBP [18]) proofs: one derives contradictions by summing XOR equations recovered from the CNF encoding; the other combines exactly-one constraints and conditional equalities to show that incompatible boundary assignments violate count conservation.

2.3 Public Interface for Programming Strategies

We define a strategy as a program that selects and schedules solver operations based on the current solver state in order to decide the satisfiability of a formula. The front end of TACTIX provides methods to invoke an operation or a batch thereof, and to observe the result of the invocation.

Applying operations. The front end allows the user to invoke operations supported by the TACTIX backend at various granularity. The function `rules_applicable` lists operations whose requirements on the current solver state hold: for example, BVE requires decision level zero and decide requires there is no unpropagated unit. The `configure` method can be used to specify a batch of the same operation (e.g., BVE, vivify, local search) with particular settings and effort budgets. The concrete settings depend on the operation: for BVE, one can set the increase in the number of clauses caused by each elimination and a pre-defined heuristic for prioritizing variables to eliminate. Similarly, vivification allows selection of a tier of learned clauses, and local search accepts a random seed. Effort measures also vary: unit propagation is measured by propagation count, while local search charges work for updating clauses as variables are flipped. During the execution, the user can configure the same operation with different settings and limits. After the `configure` method is called, the backend validates that this is an applicable operation and registers the request. The `advance` method executes the currently configured operation repeatedly

until either the configured budgets are exhausted or a stopping point is reached (e.g., a conflict is detected). A strategy programmed over the front-end can involve invoking `configure`, one or more invocations of `advance`, and another `configure`. In addition to applying a batch of operations, the front end also allows applying a single step of a certain operation (e.g., eliminating a single variable, deciding on a single variable) with the `apply_instance` method, where the meaning of the target is determined by the applied operation. For BVE, the target is the variable to eliminate; for a decision operation, it is a literal.⁶

Observing operation effects and solver state. Each call to `advance` reports whether the operation made progress, why it stopped, and how much work it performed in a result record. Reasons to stop include completion, exhaustion of the allowance, inapplicability, conflicts, and final SAT or UNSAT answers. The function `stat` lets a strategy inspect named measurements maintained by the solver. The `statistics_id` argument shown in Figure 1 selects which quantity to read, such as the total number of conflicts, variables eliminated, or resolution attempts. It also reports current state, such as the decision level and the number of unassigned variables. Reading an accumulated total before and after an operation gives the amount attributable to that operation. For example, the user-defined strategy can compare variables eliminated with resolution attempts during a batch of BVE operations, then decide whether to request further eliminations, relax its clause-growth bound, or try another technique.⁷ The front end also allows one to inspect results after an individual operation. When an effort limit is reached, the back end will finish any transformation in progress, including its proof and model-reconstruction updates, before returning control, so that subsequent operations can proceed from a consistent state.

Programming a new strategy. Using the front-end methods, one can program strategies ranging from complete search procedures to specialized simplification sequences. The sequence of operations allowed by the front end can be represented with the following pattern:

```
(simplification | structural reasoning | local search)* ;
(search ; restart-to-root ;
 (simplification | structural reasoning | local search)*)*
```

Here, `;` denotes sequential composition, `|` denotes choice, and `*` denotes zero or more repetitions. In words, TACTIX allows arbitrary sequences of simplification, structural reasoning, and local search at decision-level zero, interleaved with CDCL search sessions. The same operation can be invoked with different settings and efforts within the same execution. Internally, the solver backend may need to initialize new heuristics required by the selected operations during execution. The default strategy of TACTIX is programmable top of the public interface. On the other hand, for a specific circuit encoding, a user-defined strategy might alternate congruence reasoning, which discovers equivalent

⁶ The documented public interface in `include/tactix.h` lists settings, effort limits, target selections, and result types.

⁷ The `TACTIX_STAT_*` identifiers in `include/tactix.h` list the available measurements.

signals, with substitution, which replaces equivalent literals. The strategy could use `stat` to measure newly discovered equivalences and substituted variables, and repeat the sequence while it remains productive. The user can also fall back to the default strategy by invoking `default_solve` after performing the sequence they designed. The back end will continue the solving process and in the end either construct a model to the original formula or produce a uniform UNSAT proof. Section 4 describes and evaluates two alternative strategies for two specific application domains that resulted in substantial performance gains.

3 Solver Development Methodology

In this project, we used AI coding agents (Claude Code and Codex) to help develop TACTIX. Figure 2 summarizes our AI-assisted solver development workflow. The developers begin with a specification covering three aspects: the solver architecture, the performance milestones that guide development (Section 3.2), and a testing harness for evaluating correctness and performance (Sections 3.1 and 3.2).

In practice, the specification is stored as a development plan document that record the system design and the non-negotiable requirements, a progress document that keeps track of the milestones, and a test harness that contains scripts and tools that the AI cannot modify without permission. The agents use the specification to implement and refine the solver. They also have access to reference implementations, including MINISAT, GLUCOSE, KISSAT, CRYPTO MINISAT, CADICAL, LYMPHO SAT, and SATSUMA+KISSAT, to study established techniques, how they are efficiently implemented, and investigate differences in solver behavior. The harness evaluates each candidate implementation and reports the results to both the developers and the agents. Together, they use this feedback to investigate correctness failures and performance bottlenecks. The developers and the AI agents are constantly interacting in this process, where the developers ask questions, provide feedback, and give concrete instructions, and the agents report progress, ask questions, identify bottlenecks, and make suggestions. The developers need to continuously revise, concretize, and amend the specification, as well as inspecting the implementation to ensure their consistency with the specified architectural design.

3.1 Correctness testing

The flexibility provided by TACTIX requires ensuring that its techniques work correctly under different user-defined strategies. A strategy may invoke techniques in orders that do not occur in the default strategy, or repeat a technique with different settings and effort limits. Testing individual techniques is therefore insufficient: we must also test their interactions under different strategies. To address this challenge, we design an evaluation for both formulas and strategies, following established SAT fuzzing approaches, including `cnffuzz` and CADICAL’s `Mobical` [12, 19], and model-based testing [6]. The harness contains a script to generate random operation sequences that we expect to support (as describe in Section 2) by repeatedly invoking operations and varying their settings and effort limits. We also design dedicated stress strategies to exercise techniques that random strategies rarely reach, using counters to confirm that the intended transformations occur.

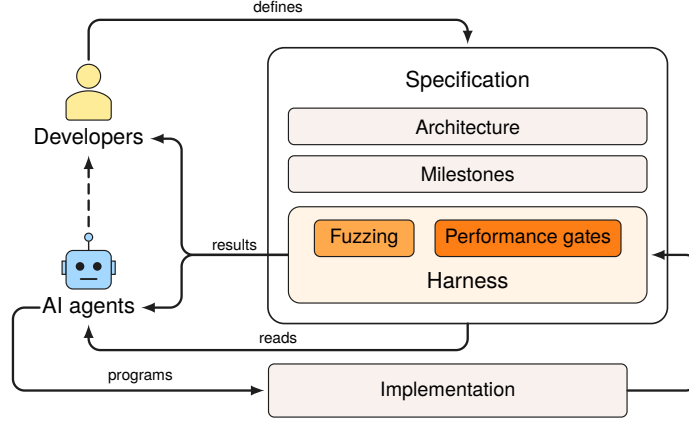


Fig. 2. The development workflow of TACTIX. Correctness and performance tests provide feedback for revising the implementation and its specification.

In practice, the random policy fails fuzzing much more often than the default policy. And when a test fails, we reduce both the formula and the operation sequence to obtain a small regression test that preserves the failure. One interesting failure cases occurred when a random strategy requested a decision before unit propagation converged. In this situation, the implementation could assign implied literals at the wrong decision level and later discard them incorrectly during backtracking. The default strategy did not expose this error because it always completed propagation before making a decision. In this case, we instead corrected the applicability check to reject decision requests while propagation is pending. To validate solver results, the harness independently checks satisfying assignments against the original formula and verifies UNSAT proofs using off-the-shelf external checkers. It is part of the specification that the solver must be able to produce proofs for any refutation operation it implements. The availability of trusted external proof checkers is *crucial* for maintaining the correctness of the implementation and catching soundness issues in time.

3.2 Performance testing and milestones

Alongside the separation of strategies from proof rules, achieving state-of-the-art performance with a default strategy is also a non-negotiable development goal. Therefore the harness also conduct performance testing on real benchmarks. The annual SAT Competitions [11, 22, 32, 59] provide benchmark collections and comparative solver evaluations for performance testing. The benchmark must be selected thoughtfully: easy instances may fail to expose bottlenecks, while instances that consistently time out provide little feedback on incremental improvements and prolong the development process. Throughout the development process, we defined six milestones (Table 1) and designed different tiers of benchmark sets (Table 2) for performance testing.

Table 1. Development milestones, performance gates, and retained benchmark sets for regression testing.

Milestone	Performance Gate	Regression Tests
1	Each Tier A instance solved within 30 s	None
2	Half of the Tier B instances solved within 300 s	Tier A
3	$\text{PAR-2} \leq 1.10 \times \text{KISSAT}$ on Tiers B and C	Tiers A and B
4	Each Tier R instance solved within 1200 s; aggregate runtime $\leq 1.10 \times \text{KISSAT}$	B28+C5
5	Each Tier W instance solved within 1200 s	B28+C5 and R
6	Selected benchmarks with GF(2) solved within 600 s	B28+C5, R, W

Table 2. Benchmark tiers used during development.

Tier	#	Use	Source
A	200	milestone 1	simple crafted benchmarks
B	28	milestone 2, regression	SC’23-24, took KISSAT 5-30 seconds
C	30	milestone 3, regression	SC’23-24, took KISSAT 5-300 seconds
R	50	milestone 4, regression	SC’24-26, hard for TACTIX, easy for KISSAT
W	100	milestone 5, regression	SC’24-26, uniquely solved by one baseline
D	1,166	final eval.	SC24–26 development milestones

Development milestones. Table 1 summarizes the performance gates and regression tests used for each milestone. The techniques and the gates might be adjusted during the development process and the table shows the final ones when the developer declares the milestone reached. The regression tests B28+C5 consist of all of the Tier B benchmarks and five of the Tier C benchmarks where the performance tends to be sensitive to implementation changes. Each milestone and its regression tests were set dynamically after the preceding milestone was reached. Section 5 evaluates the archived versions associated with these milestones on SAT Competition benchmarks.

4 Evaluation

In this section, we present an experimental evaluation that aims to answer the following questions. First, we investigate whether TACTIX can achieve state-of-the-art performance on SAT Competition benchmarks. We will show that the answer is positive. Second, we evaluate how much overhead the public interface incurs by comparing two implementations of the default strategies, one over the public interface, the other one directly over the C++ back end. We found that the overhead in this case is negligible. Third, we test the flexibility of TACTIX on two specialized sets of benchmarks that are known to be hard for modern SAT solvers. Again, the result is encouraging: we are able to find domain-specific strategies that significantly improve upon state-of-the-art solvers on

Table 3. Results on the SC’24-26 benchmark sets (400 each) and their union (1166). Solved, SAT, and UNSAT count solver answers; parenthesized “checked” columns count independently verified solutions.

Set	Configuration	Solved	Solved (checked)	SAT	UNSAT	UNSAT (checked)	Average runtime on checked solves (s)	Average PAR-2 (checked) (s)
SC24	TACTIX	355	350	167	188	183	331.1	1540
	SATSUMA+KISSAT	346	339	163	183	176	343.8	1816
	KISSAT 4.0.4	326	324	159	167	165	352.4	2185
	LYMPHOSAT	351	328	169	182	168	269.6	2021
	KISSAT-sc2024	327	326	158	169	168	371.3	2153
	AE-KISSAT-MAB	330	325	165	165	160	368.6	2174
SC25	TACTIX	368	366	173	195	193	380.3	1198
	SATSUMA+KISSAT	357	347	166	191	181	394.2	1667
	KISSAT 4.0.4	336	330	168	168	162	490.6	2155
	LYMPHOSAT	355	293	172	183	174	339.9	2924
	KISSAT-sc2024	334	328	166	168	162	488.9	2201
	AE-KISSAT-MAB	341	335	176	165	159	460.4	2011
SC26	TACTIX	313	312	158	155	154	633.8	2694
	SATSUMA+KISSAT	288	286	135	153	151	665.9	3326
	KISSAT 4.0.4	243	242	131	112	111	687.0	4366
	LYMPHOSAT	271	258	155	116	111	701.4	4002
	KISSAT-sc2024	253	253	140	113	113	788.6	4174
	AE-KISSAT-MAB	229	228	136	93	92	593.8	4638
All	TACTIX	1006	999	481	525	518	441.1	1810
	SATSUMA+KISSAT	965	946	449	516	497	459.6	2260
	KISSAT 4.0.4	881	872	444	437	428	495.0	2892
	LYMPHOSAT	947	852	478	469	442	424.0	3003
	KISSAT-sc2024	890	883	450	440	433	532.0	2830
	AE-KISSAT-MAB	875	863	462	413	401	461.5	2940

those benchmarks. All experiments were performed on a cluster equipped with AMD EPYC 7513 processors. Per-instance results for these evaluations and the development versions appear in Appendix C.

4.1 Performance of TACTIX on SAT Competition Benchmarks

We first evaluate the default strategy of TACTIX on SAT Competition (SC) 2024, 2025, and 2026 main-track benchmarks [22, 32, 59]. We compare TACTIX with competition winners of the sequential track in the past three years—KISSAT-sc2024, AE-KISSAT-MAB, and SATSUMA+KISSAT—and two additional sequential solvers—KISSAT 4.0.4 and LYMPHOSAT. Each solver run was given 4 physical cores (even though each solver spawns one thread), a CPU time limit of 5000 seconds, and 32 GB of memory. For an UNSAT answer, we checked the generated certificate using the workflow corresponding to its proof format. For DRAT certificates, we used `drat-trim` [68] to generate LRAT proofs, which were then checked by `cake_lpr` [67]. For DSR certificates, we used `dsr-trim` to generate LSR proofs, which were then checked by Trestle [23]. For PBP certificates, we used VeriPB [18] to elaborate the proofs into a format checked by `cake_pb`. For each certificate, proof elaboration and checking were performed sequentially. Each proof-checking run was given 4 physical cores, a CPU time limit of 8 hours, and 64 GB of memory.

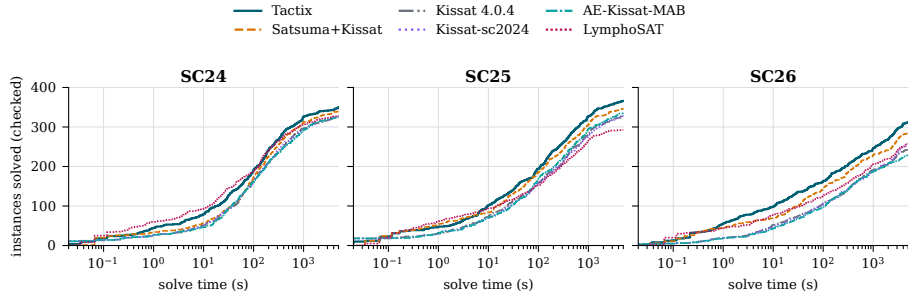


Fig. 3. Cumulative checked-solve counts on the three annual competition sets.

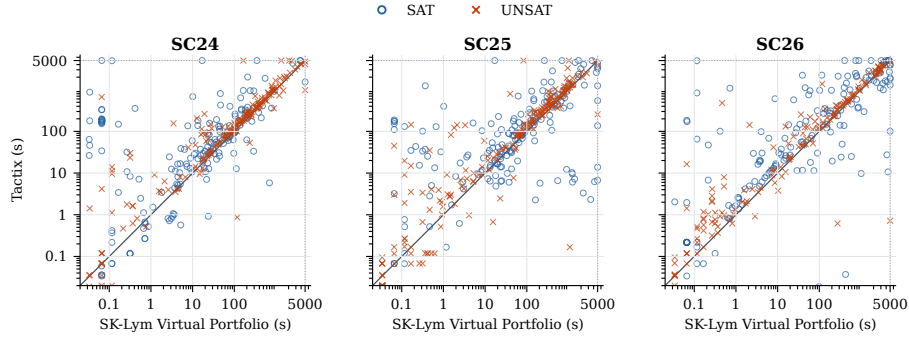


Fig. 4. TACTIX against a virtual portfolio of SATSUMA+KISSAT (SK) and LYMPHO SAT (Lym). For every formula, the portfolio uses the faster checked result of the two solvers.

Table 3 shows the performance of TACTIX across all benchmarks. Figure 3 shows the cactus plot on the same benchmarks. TACTIX outperforms the considered solvers in terms of the number of solved problems, solutions checked by the verifier, and the PAR-2 score, for each benchmark set. Figure 4 shows a comparison with a virtual portfolio of SATSUMA+KISSAT and LYMPHO SAT, which covers a large set of SAT-solving techniques. The virtual portfolio in total yields 1002 checked solutions compared to 999 for TACTIX. We see that TACTIX demonstrated complementary performance and contributed a number of unique solutions, including several very quick UNSAT solutions. Upon closer examination, those quick solutions were primarily due to the algebraic reasoning over $GF(2)$, which LYMPHO SAT only partially supports; there is also one instance which TACTIX solved immediately after the initial symmetry breaking pass but SATSUMA+KISSAT timed out on. This is due to different criteria for deciding when to perform symmetry breaking. Additional scatter plots appear in Appendix A.

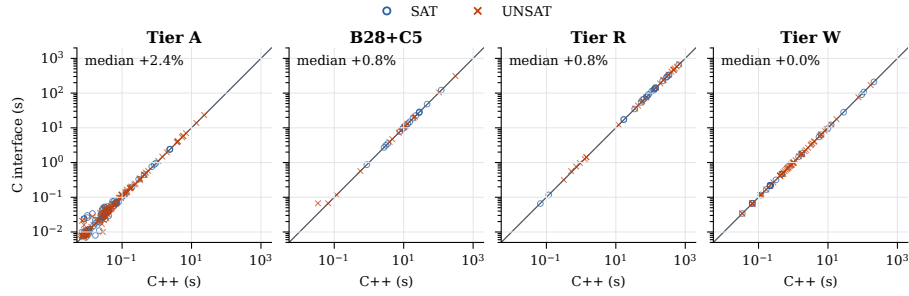


Fig. 5. Solving time of the default strategy implemented over the C interface against the same strategy implemented in C++ on four test sets.

4.2 Overhead incurred by the public interface

TACTIX comes with two versions of the default strategy: a production version implemented in C++ and is used in the experimental evaluation above, and a client version that is programmed on top of the public C interface. To compare these two versions, we evaluated them on our regular regression test suites Tiers A, B28+C5, R, and W (see Section 3) with a 1,200 s limit. Both configurations solved the same set of instances, with the same observable trace, SAT models, and UNSAT proofs.

Figure 5 compares their solving times. The median per-instance slowdown of the C interface version is 2.4% on Tier A, 0.8% on B28+C5, 0.8% on R and 0.0% on W. This suggests that the public interface of TACTIX incurs negligible overheads when it comes to implementing a complex strategy.

4.3 Designing domain-specific strategies

We now demonstrate the flexibility of TACTIX’s strategy interface by designing two domain-specific strategies for two challenging benchmark sets. The first set is a family of formulas that encode the correctness of bit-level multipliers, which is considered challenging for resolution-based solvers [31, 38, 57]. The second set is subproblems from a recent verification effort for Chvátal’s conjecture [21, 41], an important open problem in extremal set theory. In both cases, we exploited the structures of the problems to design specialized strategies over the public interface of TACTIX and show that the resulting strategies can lead to significant performance improvements.

4.4 Case study 1: arithmetic circuit verification

We first focus on a set of benchmarks on bit-level multiplier verification denoted *cruxmiter* [38, 57]. The benchmark feeds two randomly generated adder trees with the same N one-bit inputs and uses a miter to ask whether their sums can differ. The resulting formula must be unsatisfiable. We observed that the formulas have a clear circuit structure, with a large number of equivalent signals (due to the two adder trees).

Table 4. Performance of a specialized TACTIX strategy (TACTIX_{mi}) on the `cruxmiter` benchmarks.

Seed	TACTIX			TACTIX _{mi}			KISSAT 4.0.4		
	Solve (s)	Check (s)	Proof (GB)	Solve (s)	Check (s)	Proof (GB)	Solve (s)	Check (s)	Proof (GB)
0	2167.6	8071.6	4.64	823.2	1510.7	1.89	883.1	1828.0	1.96
1	2167.3	8216.7	4.64	826.9	1515.3	1.89	880.6	1829.5	1.96
2	276.3	977.1	0.70	223.9	432.2	0.54	599.2	1284.3	1.53
3	1806.1	5556.7	3.61	277.0	487.2	0.67	342.4	661.6	0.80
4	4373.0	13271.7	7.97	1386.2	3174.2	3.30	6099.0	12462.0	11.64
5	586.9	2052.3	1.47	122.0	193.9	0.31	741.2	1602.5	1.71
6	437.9	1480.0	1.05	283.6	490.5	0.70	460.5	914.8	1.14
7	196.8	599.0	0.50	41.3	64.1	0.10	115.9	198.3	0.32
8	597.3	2060.4	1.44	453.6	963.2	1.28	819.3	1872.3	1.97
9	2871.9	9527.1	5.50	421.8	802.9	0.93	5084.3	10479.6	9.25

Therefore, we instructed an AI agent to design a specialized strategy TACTIX_{mi} over the public interface of TACTIX: after propagating assignments forced at the root, it repeatedly alternates two batches of operations: congruence reasoning recognizes signals that the gate structure makes equivalent, and substitution simplifies the formula by replacing a literal with an equivalent one discovered by congruence reasoning. After each alternation, the strategy checks whether any new equivalences or substitutions were produced. If so, it repeats the process; otherwise, it applies additional inprocessing rules and hands control to a canonical CDCL search loop. In between restarts, the agent decided to only perform clause vivification because it was productive on a test instance, and disabled all other simplification and structural reasoning techniques on this family. The agent was given smaller instances of the same family (N=30) to experiment with different strategies and came up with the above strategy within 5 hours. We then evaluated the resulting strategy on the 10 largest instances of the family (N=32) and compared it with the default strategy of TACTIX and KISSAT 4.0.4. The results are shown in Table 4.

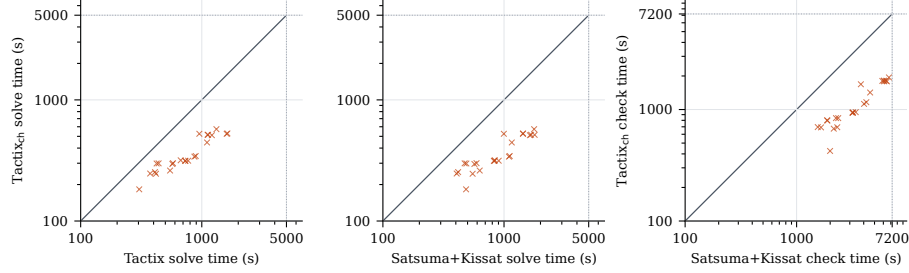
Default TACTIX and KISSAT exhibit complementary performance. On the other hand, the specialized strategy is significantly faster than both baselines. Since no symmetry-breaking was triggered, all configurations produced DRAT proof. The proofs generated by TACTIX_{mi} are smaller and faster to check than those generated by the other two solvers. On seed 4, the hardest instance, TACTIX_{mi} is 4x faster than KISSAT in terms of both solving and proof checking time. We note that it is impossible to implement this strategy in KISSAT without modifying its core architecture. This demonstrates that the strategy interface can successfully facilitate the development of novel and practically useful solving strategies.

4.5 Case study 2: Chvátal’s Conjecture

Our second case study focuses on the Chvátal’s conjecture [21], a long-standing open problem in extremal combinatorics. Recent work verifies finite cases of this conjecture by partitioning the space of candidate counterexamples into subproblems [41]. The subproblems are challenging for modern SAT solvers, as they contain a number of long cardinality constraints [9]. It was reported that solver parameter tuning, preprocessing with SATSUMA, and solving with cardinality-aware solvers [56] did not significantly

Table 5. Performance of a specialized TACTIX strategy (TACTIX_{ch}) on 23 Chvátal subproblems.

Configuration	Solved	Runtime (s)	Checked	Check time (s)
TACTIX _{ch}	23	8,463	23	26,562
TACTIX	23	18,799	23	41,357
SATSUMA+KISSAT	23	22,121	23	85,165
KISSAT 4.0.4	23	21,791	23	34,151

**Fig. 6.** TACTIX_{ch} on the 23 Chvátal subproblems. Left and middle: solving time against TACTIX and SATSUMA+KISSAT. Right: proof-checking time against SATSUMA+KISSAT.

help improve performance [41]. We consider all of the subproblems for finite case 7 that KISSAT 4.0.4 solves in 300 s to 1800 s. We gave 3 of the 23 subproblems to an AI agent and asked it to design a specialized strategy. Within 2 hours, it found that the totalizer constraints in the encoding count the same variables in the same order, and came up with a strategy, denoted TACTIX_{ch}, that applies substitution-redundancy steps to merge the duplicated counter variables. This preprocessing merges 4,508 duplicate counter variables, corresponding to 53.1% of the encoding’s auxiliary variables. After this preprocessing pass, the agent falls back to the default strategy. We evaluated TACTIX_{ch} on all 23 subproblems and compared it with the default strategy of TACTIX, KISSAT 4.0.4, and SATSUMA+KISSAT. The results are shown in Table 5 and Figure 6.

SATSUMA+KISSAT was not able to identify the symmetry and performed worse than KISSAT 4.0.4, which is consistent with what was reported in previous work [41]. On the other hand, the variable merging enabled by TACTIX_{ch} resulted in at least 2x speed up overall against any baselines. This suggests that knowledge of the encoding can guide useful simplifications, and that TACTIX enables users to implement such simplifications as specialized solving strategies.

5 Development progress and discussion

In this section, we study our AI-assisted solver development process and discuss what additional infrastructure might be needed to help accelerate it. We identified the source versions associated with the six performance milestones in Table 1. We evaluated how performance improved over the development process by running the five previous versions

Table 6. Performance and implementation activity of the evaluated versions. Days are counted from project start. Commits and added/deleted lines cover the interval since the preceding row, or project start.

Version	Commit	Day	Solved	Commits	Solver/API lines added / deleted	Test/fuzzing lines added / deleted
1	dcc3e	5	625	47	3,679 / 416	1,726 / 24
2	54629	11	735	131	9,488 / 1,599	3,286 / 225
3	6f44f	34	818	831	48,943 / 24,954	15,295 / 3,150
4	f50e4	71	977	1,003	57,939 / 13,778	37,196 / 1,782
5	0a906	80	979	246	13,282 / 1,383	8,964 / 447
Now	0f7c1	93	1,006	603	84,566 / 17,540	59,077 / 9,253

on the 1,166 SC24–26 benchmarks. Each solver was given the same computational resources: 4 physical cores, 32 GB of memory, and a 5,000 s time limit. The cumulative distributions of solving time for SAT and UNSAT instances are shown in Figure 7.

We now describe the actual implementation work in each of the versions and relate it to performance. Version 1 implemented basic CDCL search with restarts and learned-clause reduction, which already solves a majority of the benchmarks. Version 2 added inprocessing techniques including BVE, vivification, subsumption, and congruence. Version 3 added local search, SAT sweeping, and blocked and covered clause elimination. Both versions resulted in substantial net gains in solved instances. The largest increase came in the 37 days from version 3 to version 4. Version 4 improved variable-elimination and congruence implementations and integrated structured symmetry preprocessing and native substitution-redundancy proofs. This version newly solved 183 formulas and lost 24, for a net gain of 159. Of the newly solved formulas, 134 were UNSAT. Milestone 5 shifted attention to a fixed set of 100 formulas for which exactly one of the baselines had a checked solve within 1200 s. It turned out that most were covered by SATSUMA+KISSAT or LYMPHO SAT. To reach the milestone, version 5 incorporated several structural reasoning techniques from LYMPHO SAT as local search passes and improved the proof-checking pipeline. Reaching the target took nine days. Interestingly, across the full competition benchmark set, this version achieved a net gain of 24 SAT instances and a net loss of 22 UNSAT instances. This decrease in UNSAT performance was due to an attempt to warm-start CDCL search with partial assignments from the newly introduced local search passes, which turned out to hurt performance. The following 13 days added algebraic reasoning from CRYPTO MINISAT and additional structural reasoning inspired by LYMPHO SAT. These additions recovered a number of UNSAT instances lost at the previous milestone and resulted in the current version.

Table 6 also records repository activity between versions. Before milestone 3, substantial changes were concentrated in the default strategy, solver core, and simplification techniques. Between milestones 3 and 4, the work extended to symmetry reasoning, configurable strategies, and diagnostic tools for comparing search behavior. The final interval includes extensive structural reasoning and public-interface work, together with 59,077 added and 9,253 deleted lines of test and fuzzing code. It also includes 16,144

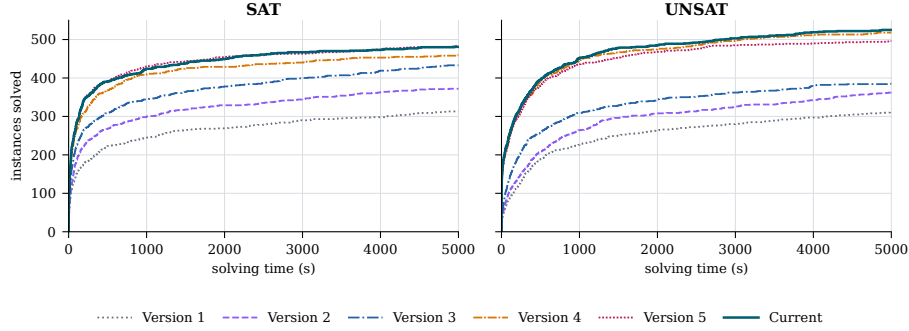


Fig. 7. Performance of the evaluated versions on SAT and UNSAT benchmarks from SC24–26.

added and 1,034 deleted lines in reusable experiment tools. These changes supported correctness, testing, and programmability alongside performance improvements.

Testing and fuzzing also accounted for a substantial share of implementation activity. Counting both added and deleted lines in Table 6, changes to test and fuzzing code amounted to approximately 51% of changes to solver and API code over the full development period. This ratio varied across intervals, rising from 25% in the interval leading to version 3 to 54%, 64%, and 67% in the three subsequent intervals. The later stages involved an increasing amount of test and fuzzing code change relative to solver code change. This is because correctness testing becomes more challenging as new techniques and strategy interactions are introduced (Section 3.1).

Infrastructure for AI-assisted solver development. While we believe the use of AI agents accelerated our solver development process, we still encountered a number of bottlenecks. To ensure the correctness of our implementation, whenever we committed changes that affected solver behavior, we performed extensive fuzzing of both random strategies and the default strategy, in both production and debugging modes, and ran regression tests. In this process, we checked each generated UNSAT proof, which created significant overhead: checking an UNSAT proof for a benchmark solved in 10 minutes sometimes took up to an hour. Moreover, proof certificates can be large and create storage pressure if they were not cleaned up promptly. We believe *online proof checking* [60] is a promising approach to tackle this bottleneck and accelerate the development cycle. In addition, fixing the performance gate to a small set of benchmarks risks overfitting to them; this was particularly evident in the transition from milestone 4 to milestone 5. It could be interesting to develop *performance fuzzers*, which might construct synthetic formulas by combining specific structures to stress-test particular techniques. This could enable us to evaluate solver performance in a more targeted manner and potentially uncover performance issues much more efficiently.

6 Related Work

Modern SAT solvers provide several ways to influence solving. `CADICAL` offers configuration options, incremental solving, and interfaces for external propagation and proof tracing [12]. In particular, `IPASIR-UP` allows an external program to propagate literals, supply reasons, and influence branching decisions [26]. These interfaces support close interaction with a running solver. Automated parameter tuning can substantially improve SAT solver performance [35]. `TACTIX` broadens the space of expressible strategies by allowing clients to select, configure, and interleave search and inprocessing operations during execution. This interface supports experimentation with new solving strategies using the reasoning techniques and proof systems implemented by the backend. The separation of reasoning rules from their application strategies is well established in abstract SAT and SMT calculi [37, 51]. Marić’s formalization, which underlies `ARGOSAT`, separates heuristic choices from the algorithms whose correctness is proved [46]. `TACTIX` builds on this principle by exposing the techniques of a proof-producing inprocessing CDCL solver through a public strategy interface. Strategy languages in SMT solvers, such as `Z3`, allow users to compose simplification and solving procedures into tactics [49]. `TACTIX` is designed to expose similar flexibility within a sequential CDCL solver. `TACTIX` uses established proof formats and checking tools. `DRAT` and `LRAT` support clausal proof production and efficient checking [24, 68]; substitution redundancy supports transformations such as symmetry breaking [20, 23]. Our testing infrastructure also builds on formula fuzzing and model-based API testing [6, 19], extending them to varied and randomly generated sequences of solver operations.

7 Conclusion

We introduced `TACTIX`, which is both a high-performing proof-producing CDCL solver and a library that allows users to program novel solving strategies. `TACTIX` achieves this by separating the solving techniques from the strategies that orchestrate them. `TACTIX` performs competitively against other state-of-the-art solvers on benchmarks from the past three editions of the SAT Competition. In two different SAT-solving applications, we showed that one can use `TACTIX` to define solving strategies that deviate significantly from those in existing SAT solvers and lead to substantial performance gains. We also reported on our AI-assisted solver development process and discussed potential directions for infrastructure development in the formal methods community. In addition to supporting additional operations [61, 64], `TACTIX` provides infrastructure for systematically evaluating and developing SAT-solving strategies based on online learning, a vision shared by several existing lines of work [28, 62, 69]. Finally, it would be interesting to develop a domain-specific language on top of `TACTIX`’s API to make it more convenient to synthesize domain-specific SAT solvers.

AI Use Statement Claude Code and OpenAI Codex were used to assist with implementation, testing, debugging, and manuscript preparation, including language editing and plot generation. Section 3 describes the human-supervised AI-assisted solver development process. The authors are responsible for the content and conclusions of this paper.

Declaration of Competing Interests The authors declare no competing interests.

References

1. Anders, M., Brenner, S., Rattan, G.: Satsuma: Structure-based symmetry breaking in SAT. In: 27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024). LIPIcs, vol. 305, pp. 4:1–4:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2024). <https://doi.org/10.4230/LIPIcs.SAT.2024.4>
2. Anders, M., Codel, C., Heule, M.J.H.: Orbitopal fixing in SAT. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2026), Part I. LNCS, vol. 16505, pp. 89–109. Springer (2026). https://doi.org/10.1007/978-3-032-22752-2_5
3. Anders, M., Codel, C., Heule, M.J.H.: Simplify, order, break, repeat. In: 29th International Conference on Theory and Applications of Satisfiability Testing (SAT 2026). LIPIcs, vol. 377, pp. 4:1–4:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2026). <https://doi.org/10.4230/LIPIcs.SAT.2026.4>
4. Anders, M., Schweitzer, P.: Parallel computation of combinatorial symmetries. In: 29th Annual European Symposium on Algorithms (ESA 2021). LIPIcs, vol. 204, pp. 6:1–6:18 (2021). <https://doi.org/10.4230/LIPIcs.ESA.2021.6>
5. Andersson, G., Bjesse, P., Cook, B., Hanna, Z.: A proof engine approach to solving combinatorial design automation problems. In: Design Automation Conference (DAC 2002). pp. 725–730 (2002). <https://doi.org/10.1145/514100.514101>
6. Artho, C., Biere, A., Seidl, M.: Model-based testing for verification back-ends. In: Tests and Proofs (TAP 2013). LNCS, vol. 7942, pp. 39–55. Springer (2013). https://doi.org/10.1007/978-3-642-38916-0_3
7. Audemard, G., Simon, L.: Predicting learnt clauses quality in modern SAT solvers. In: International Joint Conference on Artificial Intelligence (IJCAI 2009). pp. 399–404 (2009). <https://www.ijcai.org/Proceedings/09/Papers/074.pdf>
8. Bacchus, F., Winter, J.: Effective preprocessing with hyper-resolution and equality reduction. In: Theory and Applications of Satisfiability Testing (SAT 2003). LNCS, vol. 2919, pp. 341–355 (2004). https://doi.org/10.1007/978-3-540-24605-3_26
9. Bailleux, O., Bouffkhad, Y.: Efficient cnf encoding of boolean cardinality constraints. In: Rossi, F. (ed.) Principles and Practice of Constraint Programming – CP 2003. pp. 108–122. Springer Berlin Heidelberg, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45193-8_8
10. Balint, A., Schöning, U.: Choosing probability distributions for stochastic local search and the role of make versus break. In: Theory and Applications of Satisfiability Testing (SAT 2012). LNCS, vol. 7317, pp. 16–29 (2012). https://doi.org/10.1007/978-3-642-31612-8_3
11. Balyo, T., Heule, M.J.H., Iser, M., Jarvisalo, M., Suda, M. (eds.): Proceedings of SAT Competition 2023: Solver, Benchmark and Proof Checker Descriptions, Department of Computer Science Series of Publications B, vol. B-2023-1. Department of Computer Science, University of Helsinki (2023), <https://hdl.handle.net/10138/563824>
12. Biere, A., Faller, T., Fazekas, K., Fleury, M., Froleys, N., Pollitt, F.: CaDiCaL 2.0. In: Computer Aided Verification (CAV 2024). LNCS, vol. 14681, pp. 133–152. Springer (2024). https://doi.org/10.1007/978-3-031-65627-9_7
13. Biere, A., Faller, T., Fazekas, K., Fleury, M., Froleys, N., Pollitt, F.: CaDiCaL, Gimsatul, IsaSAT and Kissat entering the SAT Competition 2024. In: Proceedings of SAT Competition 2024: Solver, Benchmark and Proof Checker Descriptions. Department of Computer Science Report Series B, vol. B-2024-1, pp. 8–10. University of Helsinki (2024), <https://cca.informatik.uni-freiburg.de/papers/BiereFallerFazekasFleuryFroleysPollitt-SAT-Competition-2024-solvers.pdf>

14. Biere, A., Fazekas, K., Fleury, M., Froleys, N.: Clausal congruence closure. In: 27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024). LIPIcs, vol. 305, pp. 6:1–6:25 (2024). <https://doi.org/10.4230/LIPIcs.SAT.2024.6>
15. Biere, A., Fazekas, K., Fleury, M., Heisinger, M.: CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In: Proceedings of SAT Competition 2020: Solver and Benchmark Descriptions. Department of Computer Science Report Series B, vol. B-2020-1, pp. 51–53. University of Helsinki (2020), <https://fmv.jku.at/papers/BiereFazekasFleuryHeisinger-SAT-Competition-2020-solvers.pdf>
16. Biere, A., Fleury, M., Heisinger, M.: CaDiCaL, Kissat, Paracooba entering the SAT Competition 2021. In: Proceedings of SAT Competition 2021: Solver and Benchmark Descriptions. Department of Computer Science Report Series B, vol. B-2021-1, pp. 10–13. University of Helsinki (2021), <http://hdl.handle.net/10138/333647>
17. Biere, A., Fröhlich, A.: Evaluating CDCL variable scoring schemes. In: Theory and Applications of Satisfiability Testing (SAT 2015). LNCS, vol. 9340, pp. 405–422 (2015). https://doi.org/10.1007/978-3-319-24318-4_29
18. Bogaerts, B., Gocht, S., McCreesh, C., Nordström, J.: Certified dominance and symmetry breaking for combinatorial optimisation. *Journal of Artificial Intelligence Research* **77**, 1539–1589 (2023). <https://doi.org/10.1613/jair.1.14296>
19. Brummayer, R., Lonsing, F., Biere, A.: Automated testing and debugging of SAT and QBF solvers. In: Theory and Applications of Satisfiability Testing (SAT 2010). LNCS, vol. 6175, pp. 44–57. Springer (2010). https://doi.org/10.1007/978-3-642-14186-7_6
20. Buss, S., Thapen, N.: DRAT and propagation redundancy proofs without new variables. *Logical Methods in Computer Science* **17**(2), 12:1–12:31 (2021). [https://doi.org/10.23638/LMCS-17\(2:12\)2021](https://doi.org/10.23638/LMCS-17(2:12)2021)
21. Chvátal, V.: Intersecting families of edges in hypergraphs having the hereditary property. In: Berge, C., Ray-Chaudhuri, D. (eds.) *Hypergraph Seminar*. pp. 61–66. Springer Berlin Heidelberg, Berlin, Heidelberg (1974). <https://doi.org/10.1007/bfb0066179>
22. Codel, C., Fazekas, K., Heule, M.J.H., Iser, M. (eds.): *Proceedings of SAT Competition 2025: Solver and Benchmark Descriptions*. Proceedings of SAT Competitions, TU Wien Academic Press (2025). <https://doi.org/10.34726/10379>, <https://repositum.tuwien.at/handle/20.500.12708/218424>
23. Codel, C.R., Avigad, J., Heule, M.J.H.: Verified substitution redundancy checking. In: *Formal Methods in Computer-Aided Design (FMCAD 2024)*. pp. 186–196. TU Wien Academic Press (2024). https://doi.org/10.34727/2024/isbn.978-3-85448-065-5_24
24. Cruz-Filipe, L., Heule, M.J.H., Hunt, W.A., Kaufmann, M., Schneider-Kamp, P.: Efficient certified RAT verification. In: *Automated Deduction (CADE 2017)*. LNCS, vol. 10395, pp. 220–236. Springer (2017). https://doi.org/10.1007/978-3-319-63046-5_14
25. Eén, N., Biere, A.: Effective preprocessing in SAT through variable and clause elimination. In: *Theory and Applications of Satisfiability Testing (SAT 2005)*. LNCS, vol. 3569, pp. 61–75. Springer (2005). https://doi.org/10.1007/11499107_5
26. Fazekas, K., Niemetz, A., Preiner, M., Kirchweger, M., Szeider, S., Biere, A.: IPASIR-UP: User propagators for CDCL. In: 26th International Conference on Theory and Applications of Satisfiability Testing (SAT 2023). LIPIcs, vol. 271, pp. 8:1–8:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2023). <https://doi.org/10.4230/LIPIcs.SAT.2023.8>
27. Fleury, M., Biere, A.: Efficient all-UIP learned clause minimization. In: *Theory and Applications of Satisfiability Testing (SAT 2021)*. LNCS, vol. 12831, pp. 171–187 (2021). https://doi.org/10.1007/978-3-030-80223-3_12
28. Ganesh, V., Seshia, S.A., Jha, S.: Machine learning and logic: A new frontier in artificial intelligence. *Formal Methods in System Design* **60**(3), 426–451 (2022). <https://doi.org/10.1007/s10703-023-00430-1>

29. Green, H., Le Goues, C., Brown, F.: The case for automated hyperspecialization: Evidence from SAT (2026). <https://doi.org/10.48550/arXiv.2609.14836>, <https://arxiv.org/abs/2609.14836>
30. Han, H., Somenzi, F.: On-the-fly clause improvement. In: Theory and Applications of Satisfiability Testing (SAT 2009). LNCS, vol. 5584, pp. 209–222 (2009). https://doi.org/10.1007/978-3-642-02777-2_21
31. Heisinger, M., Fleury, M., Biere, A.: Distributed cube and conquer with paracooba. In: International Conference on Theory and Applications of Satisfiability Testing. pp. 114–122. Springer (2020). https://doi.org/10.1007/978-3-030-51825-7_9
32. Heule, M.J.H., Iser, M., Järvisalo, M., Suda, M. (eds.): Proceedings of SAT Competition 2024: Solver, Benchmark and Proof Checker Descriptions, Department of Computer Science Report Series B, vol. B-2024-1. Department of Computer Science, University of Helsinki (2024), <https://hdl.handle.net/10138/584822>
33. Heule, M.J.H., Järvisalo, M., Biere, A.: Efficient CNF simplification based on binary implication graphs. In: Theory and Applications of Satisfiability Testing (SAT 2011). LNCS, vol. 6695, pp. 201–215 (2011). https://doi.org/10.1007/978-3-642-21581-0_17
34. Heule, M.J.H., Järvisalo, M., Biere, A.: Covered clause elimination. In: LPAR-17-short: Short Papers for the 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning. EPIc Series in Computing, vol. 13, pp. 41–46. EasyChair (2013). <https://doi.org/10.29007/cl8s>, conference held in 2010
35. Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T.: ParamILS: An automatic algorithm configuration framework. Journal of Artificial Intelligence Research **36**, 267–306 (2009). <https://doi.org/10.1613/jair.2861>
36. Järvisalo, M., Biere, A., Heule, M.J.H.: Blocked clause elimination. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010). LNCS, vol. 6015, pp. 129–144 (2010). https://doi.org/10.1007/978-3-642-12002-2_10
37. Järvisalo, M., Heule, M.J.H., Biere, A.: Inprocessing rules. In: Automated Reasoning (IJCAR 2012). LNCS, vol. 7364, pp. 355–370. Springer (2012). https://doi.org/10.1007/978-3-642-31365-3_28
38. Kaufmann, D., Kauers, M., Biere, A., Cok, D.: Arithmetic verification problems submitted to the SAT Race 2019. In: Proceedings of SAT Race 2019: Solver and Benchmark Descriptions. Department of Computer Science Report Series B, vol. B-2019-1, p. 49. University of Helsinki (2019), <https://danielakaufmann.at/wp-content/uploads/2020/11/KaufmannKauersBiereCok-SAT-Race-2019-benchmarks.pdf>
39. Kiesl, B., Heule, M.J.H., Biere, A.: Truth assignments as conditional autarkies. In: Automated Technology for Verification and Analysis (ATVA 2019). LNCS, vol. 11781, pp. 48–64 (2019). https://doi.org/10.1007/978-3-030-31784-3_3
40. Kuehlmann, A., Paruthi, V., Krohm, F., Ganai, M.K.: Robust Boolean reasoning for equivalence checking and functional property verification. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **21**(12), 1377–1394 (2002). <https://doi.org/10.1109/tcad.2002.804386>
41. Looney, J., McDonald, J., Klingler, A., Wu, G., Pulaj, J., Wu, H.: Solution space partitioning for extremal set theory. In: Proceedings of the 26th Conference on Formal Methods in Computer-Aided Design (FMCAD 2026). pp. 757–768. TU Wien Academic Press (2026). https://doi.org/10.34727/2026/isbn.978-3-85448-093-8_75, <https://repositum.tuwien.at/handle/20.500.12708/230570>
42. Luby, M., Sinclair, A., Zuckerman, D.: Optimal speedup of Las Vegas algorithms. Information Processing Letters **47**(4), 173–180 (1993). [https://doi.org/10.1016/0020-0190\(93\)90029-9](https://doi.org/10.1016/0020-0190(93)90029-9)

43. Luo, M., Li, C.M., Xiao, F., Manyà, F., Lü, Z.: An effective learnt clause minimization approach for CDCL SAT solvers. In: International Joint Conference on Artificial Intelligence (IJCAI 2017). pp. 703–711 (2017). <https://doi.org/10.24963/ijcai.2017/98>
44. Lynce, I., Marques-Silva, J.: Probing-based preprocessing techniques for propositional satisfiability. In: IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2003). pp. 105–110 (2003). <https://doi.org/10.1109/tai.2003.1250177>
45. Manthey, N., Heule, M.J.H., Biere, A.: Automated reencoding of boolean formulas. In: Hardware and Software: Verification and Testing (HVC 2012). LNCS, vol. 7857, pp. 102–117. Springer (2013). https://doi.org/10.1007/978-3-642-39611-3_14
46. Marić, F.: Formalization and implementation of modern SAT solvers. *Journal of Automated Reasoning* **43**(1), 81–119 (2009). <https://doi.org/10.1007/s10817-009-9127-8>
47. Marques-Silva, J.P., Sakallah, K.A.: GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers* **48**(5), 506–521 (1999). <https://doi.org/10.1109/12.769433>
48. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an efficient SAT solver. In: Design Automation Conference (DAC 2001). pp. 530–535 (2001). <https://doi.org/10.1109/dac.2001.935565>
49. de Moura, L., Passmore, G.O.: The strategy challenge in SMT solving. In: Automated Reasoning and Mathematics, LNCS, vol. 7788, pp. 15–44. Springer (2013). https://doi.org/10.1007/978-3-642-36675-8_2
50. Nadel, A., Ryvchin, V.: Chronological backtracking. In: Theory and Applications of Satisfiability Testing (SAT 2018). LNCS, vol. 10929, pp. 111–121 (2018). https://doi.org/10.1007/978-3-319-94144-8_7
51. Nieuwenhuis, R., Oliveras, A., Tinelli, C.: Solving SAT and SAT modulo theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T). *Journal of the ACM* **53**(6), 937–977 (2006). <https://doi.org/10.1145/1217856.1217859>
52. Oh, C.: Between SAT and UNSAT: The fundamental difference in CDCL SAT. In: Theory and Applications of Satisfiability Testing (SAT 2015). LNCS, vol. 9340, pp. 307–323 (2015). https://doi.org/10.1007/978-3-319-24318-4_23
53. Philipp, T., Rebola-Pardo, A.: Drat proofs for xor reasoning. In: European Conference on Logics in Artificial Intelligence. pp. 415–429. Springer (2016). https://doi.org/10.1007/978-3-319-48758-8_27
54. Piette, C., Hamadi, Y., Saïs, L.: Vivifying propositional clausal formulae. In: 18th European Conference on Artificial Intelligence (ECAI 2008). *Frontiers in Artificial Intelligence and Applications*, vol. 178, pp. 525–529. IOS Press (2008). <https://doi.org/10.3233/978-1-58603-891-5-525>
55. Pipatsrisawat, K., Darwiche, A.: A lightweight component caching scheme for satisfiability solvers. In: Theory and Applications of Satisfiability Testing (SAT 2007). LNCS, vol. 4501, pp. 294–299 (2007). https://doi.org/10.1007/978-3-540-72788-0_28
56. Reeves, J.E., Heule, M.J.H., Bryant, R.E.: From clauses to klauses. In: Computer Aided Verification (CAV 2024). LNCS, vol. 14681, pp. 110–132. Springer (2024). https://doi.org/10.1007/978-3-031-65627-9_6
57. Ritirc, D., Biere, A., Kauers, M.: Column-wise verification of multipliers using computer algebra. In: Formal Methods in Computer-Aided Design (FMCAD 2017). pp. 23–30. IEEE (2017). <https://doi.org/10.23919/FMCAD.2017.8102237>
58. Ryan, L.: Efficient Algorithms for Clause-Learning SAT Solvers. Master’s thesis, Simon Fraser University (2004), <https://www.cs.sfu.ca/~mitchell/grads.html>
59. SAT Competition Organizers: SAT Competition 2026. <https://satcompetition.github.io/2026/> (2026), accessed 16 October 2026

60. Schreiber, D.: Trusted scalable SAT solving with on-the-fly LRAT checking. In: 27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024). LIPIcs, vol. 305, pp. 25:1–25:19 (2024). <https://doi.org/10.4230/LIPIcs.SAT.2024.25>
61. Shah, A., Byrnes, T., Reeves, J., Heule, M.J.H.: Learning short clauses via conditional autarkies. In: Formal Methods in Computer-Aided Design (FMCAD 2025). pp. 115–125. TU Wien Academic Press (2025). https://doi.org/10.34727/2025/isbn.978-3-85448-084-6_17
62. Shi, Z., Jiang, W., Zhang, X., Luo, J., Liang, Y., Chu, Z., Xu, Q.: DynamicSAT: Dynamic configuration tuning for SAT solving. In: 31st International Conference on Principles and Practice of Constraint Programming (CP 2025). Leibniz International Proceedings in Informatics (LIPIcs), vol. 340, pp. 34:1–34:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2025). <https://doi.org/10.4230/LIPIcs.CP.2025.34>
63. Soos, M., Gocht, S., Meel, K.S.: Tinted, detached, and lazy CNF-XOR solving and its applications to counting and sampling. In: Computer Aided Verification (CAV 2020). LNCS, vol. 12224, pp. 463–484 (2020). https://doi.org/10.1007/978-3-030-53288-8_22
64. Soos, M., Nohl, K., Castelluccia, C.: Extending SAT solvers to cryptographic problems. In: Theory and Applications of Satisfiability Testing (SAT 2009). LNCS, vol. 5584, pp. 244–257. Springer (2009). https://doi.org/10.1007/978-3-642-02777-2_24
65. Sörensson, N., Biere, A.: Minimizing learned clauses. In: Theory and Applications of Satisfiability Testing (SAT 2009). LNCS, vol. 5584, pp. 237–243 (2009). https://doi.org/10.1007/978-3-642-02777-2_23
66. van der Tak, P., Ramos, A., Heule, M.J.H.: Reusing the assignment trail in CDCL solvers. *Journal on Satisfiability, Boolean Modeling and Computation* 7(4), 133–138 (2011). <https://doi.org/10.3233/sat190082>
67. Tan, Y.K., Heule, M.J.H., Myreen, M.O.: cake_lpr: Verified propagation redundancy checking in CakeML. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2021). LNCS, vol. 12652, pp. 223–241. Springer (2021). https://doi.org/10.1007/978-3-030-72013-1_12
68. Wetzler, N., Heule, M.J.H., Hunt, Jr., W.A.: DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In: Theory and Applications of Satisfiability Testing (SAT 2014). LNCS, vol. 8561, pp. 422–429. Springer (2014). https://doi.org/10.1007/978-3-319-09284-3_31
69. Wu, H., Barrett, C., Narodytska, N.: Cubing for tuning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 14361–14370 (2026). <https://doi.org/10.1609/aaai.v40i17.38451>
70. Xu, L., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Satzilla: portfolio-based algorithm selection for sat. *Journal of artificial intelligence research* 32, 565–606 (2008). <https://doi.org/10.1613/jair.2490>

A Additional plots and results

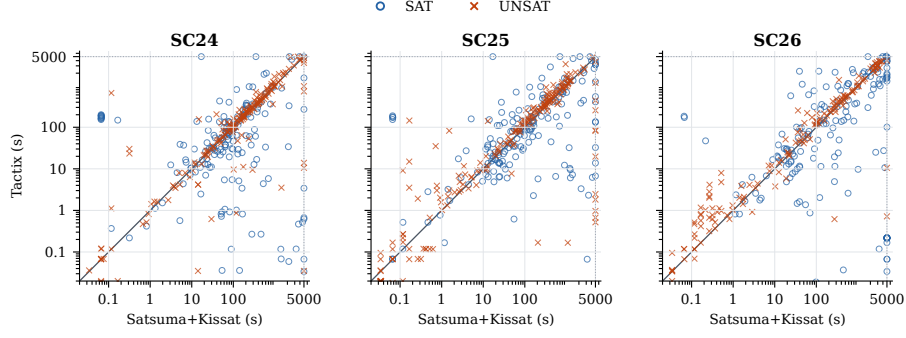


Fig. 8. Tactix against SATSUMA+KISSAT on SC24, SC25, and SC26.

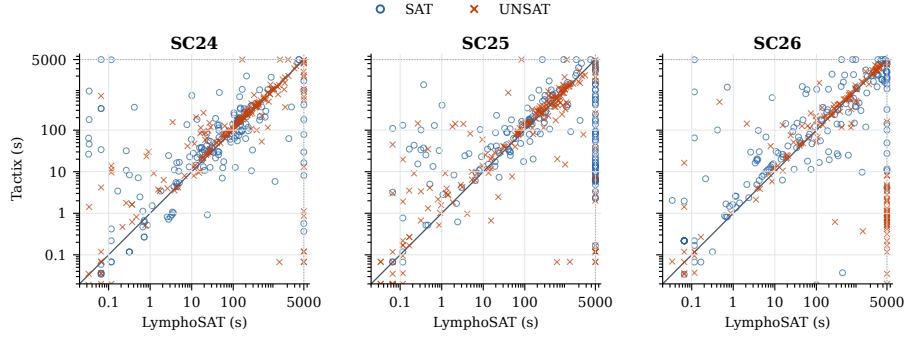


Fig. 9. Tactix against LYMPHOSAT on SC24, SC25, and SC26. Circles denote SAT and crosses UNSAT. For a result checked on only one side, the other side is plotted at the 5000 s solving-limit boundary; formulas checked by neither are omitted. Points below the diagonal favor Tactix.

Table 7. Aggregate cruxmiter results.

Configuration	Accepted	Solve total (s)	Check total (s)	Proof (GB)
TACTIX	10	15,481.213	51,812.555	31.502
TACTIX _{mi}	10	4,859.624	9,634.313	11.607
KISSAT 4.0.4	10	16,025.548	33,132.869	32.296

B Software versions

Evaluated solver versions. The Tier D evaluation uses TACTIX commit [0f7c10174faa](#). The final fuzzing campaign for the evaluated TACTIX version comprised 10,000 default-strategy cases, 10,000 random-strategy cases, 5,000 cases with graph reasoning disabled, and 5,000 cases with affine reasoning enabled. Each campaign also replayed 51 retained regression cases. No failures were detected and all completed SAT and UNSAT results passed independent model or proof checking.

KISSAT 4.0.4 was built from the [re1-4.0.4](#) tag of the [upstream repository](#); KISSAT-SC2024 uses commit [1e56df6c9bd2](#) of the same repository. The other baselines were obtained from the official SAT Competition submission archives: [AE-KISSAT-MAB \(2025\)](#), [SATSUMA+KISSAT \(2026\)](#), and [LYMPHOSAT \(2026\)](#). The SC26 submissions were rebuilt for the cluster’s processors without changing their algorithms.

Proof-checker versions. The DRAT pipeline uses `drat-trim` commit [2e3b2dc0ecf9](#) and `cake_lpr` commit [a4323b203cc9](#). The DSR pipeline uses `dsr-trim` commit [0b8cc0ef4d22](#) and `Trestle` commit [03ac9418bc23](#) with an archived bug fix. The PBP pipeline uses VeriPB 3.0.2 and `cake_pb` commit [6edb2f3585f1](#).

C Per-instance evaluation results

Filenames show the first five hash characters followed by the benchmark name. S/U denotes the reported SAT/UNSAT outcome; no runs disagree on these outcomes. TO denotes a timeout, M a solver memory limit, E1/E134 an exit without an answer, and ? an UNKNOWN result with no recorded termination reason. A dash denotes an unavailable measurement.

C.1 Competition solvers and development versions

T denotes TACTIX, K4 denotes KISSAT 4.0.4, K24 denotes KISSAT-sc2024, AE denotes AE-KISSAT-MAB, SK denotes SATSUMA+KISSAT, and L denotes LYMPHO SAT. Each competition-solver cell gives solving time / proof-checking time when the latter is available. A superscript * marks an answer not accepted by the checking pipeline; p marks a correct model not accepted under the required output protocol. V1–V5 give solving times for the development versions; these columns do not assert certificate acceptance. Now is identical to T and is shown only once. The virtual SK+L portfolio takes the smaller solving time among accepted results from those two columns. SC lists competition years; instances shared across years are listed once.

Table 8: Per-instance competition and development results (competition cells: solve/check).

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
00d1f-lec_mult_KvW_12x11.sanitized	24	U	4060/10800	3900/7550	3620/7330	TO	3930/6650*	3430/45000*	?	?	?	3880	?
00d5a-st_890_86_9_572.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
00fd8-2.normalised	25	S	26.3	12.1	11.9	11.8	33	5.57P	1300	33.1	14.5	12.1	23.9
01198-arles_thres20_p10_r7533	26	U	0.069/0.14	0.01/0.084	0.01/0.084	0.01/0.084	0.117/0.096	0.034/0.026	0.018	0.217	0.018	0.034	0.034
01d6f-lightsout_sat_25_unbounded_direct_planted250_4_sat.sanitized	26	S	0.037	416	420	3280	447	426	1350	503	1590	0.018	0.018
0205e-bp4_CSO_AM_IXA_LP.normalised	25	U	155/818	88.5/681	116/790	76.9/981	120/702	118/5440	1280	724	160	114	161
02066-mrpp_4x4#12_12	24	U	7.98/17.8	4.87/7.9	6.58/7.24	5.37/4.44	6.98/11.3	6.72/36.2	303	10.2	10.8	5.49	9.98
02654-x9-10051.sat.sanitized	24	U	145/318	110/161	118/169	181/253	120/355	120/840	630	452	121	119	178
02a46-1x1x13_1x3x6_pair_4_2_0_2_5_0.sanitized	26	S	2170	845	3350	2850	1420	786	?	4460	4790	906	2280
02e4f-REGRandom-K4-L3-Seed40.sanitized	24	U	TO	TO	TO	TO	TO	TO	?	4050	?	?	?
02f6c-linked_list_swap_contents_safety_unwind69	24	U	88/2120	62/1340	60.9/16300	63.1/17100	80.7/3300	21.2/432	301	607	102	68.3	83.9
03294-GP_216_290_40	25	S	8.68	4.62	4.47	4.47	17.8	5.93P	3110	6.68	6.73	4.92	7.93
03595-j3037_9_rggt_bm1	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
03e49-schoolt-5-8-2-4-5-10-0-1-0-2-seed2.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
03e85-gm24sparc	25	U	3.38/3.69	4.02/1.08	0.917/0.933	0.918/0.933	4.42/0.751	0.316/1.33	0.167	13.4	1.12	6.63	3.62
042cd-reconf20_50_grid10_1_6141	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
043c9-miles1500.72.sanitized	26	U	4.78/2710	TO	TO	TO	TO	0.517/2580	TO	?	?	?	1.32
04648-ramsey_3_6_19.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
04d0a-168.smt2	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
04ded-mdp-28-14-sat	24	S	0.036	49.1	274	727	53.6	0.066	621	87.1	590	0.034	0.034
04e21-mp1-Nb5T15	24	S	138	82.7	63.6	25.9	13.3	97.3	78.9	?	5.22	78.8	48.5
04f43-goldcrest-and-11	25	U	271/5860	214/4920	281/5560	228/4720	210/5280	239/6110	1860	1180	493	234	252
053e0-oisc-subrv-and-nested-15	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
05c8e-nol-11-2.sanitized	24	S	334	125	285	121	207	0.066	?	?	?	3720	123
05ed6-stb_495_168.apx_2_DC-AD	24	U	144/446	132/218	103/178	184/323	151/588	105/907	164	158	147	135	250
062f5-rook-51-0-0	25	U	1510/2940	1050/1110	1000/1480	848/1280	987/862	0.417/0.366*	1210	1240	2270	960	1240
06d2c-clcql_100_6_5.normalised	25	U	35.1/63.2	TO	TO	TO	0.717/4.5	0.466/17.8	?	?	?	33.7	34.9
0757e-vdwb_k6_n400.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
07b47-marg5x5.shuffled-as.sat03-1455	24	U	0.068/0.11	TO	TO	TO	0.066/0.094	0.066/0.036	?	?	?	0.035	0.035
07c50-ctl_4201_555_unsat_pre	24	U	803/2730	746/1590	741/1390	1050/2110	790/2750	674/10700	650	652	783	814	861
07ce9-pj2013_k9	25	U	25.5/152	24.9/74.4	9.23/30.2	8.23/30.7	23.6/179	11.3/427	?	265	34.4	32.9	39.8
07df5-cabp-V-nos6.mtx.rnd-k331.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
07e64-j3037_10_mdd_bm1	24	U	27.4/59	20.7/29.7	21.9/34.6	31.5/39.6	22.8/54.2	22.1/152	37	41.2	29.9	26.6	24.2
081f1-Break_triple_04_06.xml	25	S	0.067	0.018	0.018	0.018	0.066	0.066	0.01	0.034	0.035	0.036	0.034
0876c-mp1-kliober2017s-0500-023-t12	24/25	S	30.7	12.5	40	39.5	92.4	42.8	105	15.3	16.6	18.7	52.9
089f9-simon-r18-0.sanitized	24	S	161	0.01	0.009	0.009	0.066	131	?	0.018	0.019	0.066	153
08be2-g2-hwmc15deep-bob12s02-k16	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
08ccc-simon-r16-1.sanitized	24	S	148	0.009	0.009	0.009	0.166	122	?	0.018	0.018	0.066	140
08cf5-GreenTao_2-3-5_528	26	U	143/377	132/233	141/224	208/337	159/466	703/946	174	226	184	131	263
08d56-cliquecoloring_n26_k7_c6	25	U	0.118/0.641	TO	TO	TO	0.166/0.316	TO	?	?	?	0.118	0.117
0922f-toughsat_22bits_0	26	S	30.7	15.2	10.5	62.7	16.9	10.7	49.3	81.2	24.6	21	68.7
093fa-128_125.sanitized	24	S	141	89.1	215	179	138	203	?	?	?	132	94.3
09637-mrpp_8x8#20_14	26	S	16.6	4.22	7.98	10.9	8.93	11.3	35.3	10.6	16.9	14.9	10.9
096e4-test_v7_r17_vr5_c1_s25451.smt2-cvc4	24	U	4740/10300	TO	TO	4610/8770	TO	TO	?	?	?	4690	?
09b61-x9-12063.sat.sanitized	24	S	2010	575	203	163	624	209	2500	2450	217	594	109
09bef-GP_300_140_20	25	S	4.88	2.52	2.52	2.47	18.1	2.17P	344	3.67	3.77	2.77	4.52
09c1b-x9-12092.sat.sanitized	24	S	233	351	102	90.8	373	105	459	1930	80.4	365	88.7
09cd8-s38417	25	U	2.17/2.91	1.17/1.03	0.717/1.53	0.767/1.48	1.42/1.5	0.817/1.88	2.37	4.37	1.87	2.17	1.77
09d7a-Break_unsat_04_03.xml	24	U	0.068/0.106	0.033/0.083	0.033/0.075	0.033/0.075	0.076/0.1	0.066/0.068	0.019	0.034	0.034	0.066	0.067
09e07-ntl-90d-37.sanitized	26	S	25	566	153	TO	2690	588	?	237	?	583	25.6
0a027-hwmc17miters-xits-iso-bobsmpfu.sanitized	24	U	1.77/2.45	0.166/0.932	0.166/0.933	0.166/0.933	2.72/1.75	0.266/3.64	?	9.23	1040	147	1.87
0a27e-nol-11-0.sanitized	24	S	329	267	525	1970	2190	0.066	?	?	1950	316	331
0a488-b19_1	25	U	593/1500	398/833	458/706	572/1010	370/1290	463/3610	?	?	1740	442	573
0a8a4-sum_of_three_cubes_42_known_representation	25	S	92.1	134	18.6	26.8	55.5	1.27P	?	?	1220	82.7	79.8
0aa22-sudoku-N30-12	25	U	377/1460	318/1110	220/1400	211/1310	340/623	232/3180	?	?	800	403	404
0ac5f-sokoban-p20.sas.cr.33	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
0ac6a-pcmax-scheduling-m13-2011-12813-UNSAT.sanitized	24	U	103/414	105/244	76.7/176	138/303	112/523	77.6/692	1350	576	125	117	144
0ae91-dislog_a14_x14_n24	25	S	708	341	655	732	747	0.116P	?	?	1630	694	728
0aea4-gus-md5-08	26	U	48.1/115	54.4/134	41.9/96.4	74.3/129	43/80.6	42.5/453	51.7	87.8	53	53.6	34.3
0bad2-af-synthesis_stb_50_100_4_sat	24	S	206	409	602	355	183	587	956	533	65.9	156	401
0be1e-or_randxor_k3_n600_m600	26	U	16.4/40.5	11.5/22.6	11.9/21.7	16.4/30.1	14.2/30.5	0.066/0.182	?	?	11.1	18.6	17.5
0c043-oddball_53_5_tto_zp.normalised	25	S	6.23	1400	1630	59.5	4210	1.37P	1690	?	?	831	5.08
0c9bc-par32-2.shuffled-as.sat03-1534	25	S	3220	1270	TO	TO	TO	TO	?	?	?	1280	?
0c9da-DivS_568_11.cnf.sanitized	26	S	29.7	18.9	18.9	17.1	18.7	2.62	22	56.1	29.4	31.3	20.1
0cab9-APROVE07-01	26	U	1480/4600	1340/2790	1170/2560	1900/2700	1480/4120	1230/16000	?	?	4310	2810	1400
0caee-lru_10.sanitized	24	S	37.4	29.9	23.5	23.3	64.5	31	34.3	61.3	27.5	24.4	51.1
0ceb0-SCPC-500-12	25	U	41.3/93.7	30.3/47.8	31.7/41.8	48.6/69.2	34.9/97.3	30.4/14	37.2	41.7	31.1	34.7	67.4
0cfe9-g2-ak128boothbg2msisc	24	S	1.12	0.266	0.266	0.266	1.32	0.667	?	0.367	0.368	0.919	1.12
0d082-mexam_23_20_2.sanitized	26	U	3.47/8.52	TO	TO	TO	3.43/8.62	TO	?	?	?	3.17	3.17
0d0fa-as-p147-151-o0-rot-rlf-68-i.sanitized	26	S	TO	TO	TO	TO	TO	TO	TO	4410	3680	?	?
0d316-linked_list_swap_contents_safety_unwind65	24	U	90.5/1860	51.7/1090	73.5/18400	173/40000*	81.4/2940	17.5/320	192	594	93.4	69	79.8
0d3fc-j3045_10_gmto_b	24	S	328	56.3	181	24.8	220	183	52	370	493	283	85.7
0d817-nol-11-20.sanitized	24	S	64.2	30	64.2	15.2	119	0.033	?	3120	67.3	67.2	63.3
0d877-pj2016_k100	25	S	714	483	345	502	810	475	?	?	2770	648	660
0dd92-pj2002_k500	25	?	M	E134	TO	TO	M	TO	?	?	?	?	?
0e068-ntl-90d-39.sanitized	26	S	TO	732	116	TO	TO	745	3410	667	357	850	?
0e1d5-Break_12_50.xml	25	S	4.17	1.82	2.77	3.12	4.43	3.02	51.7	8.88	4.67	1.32	3.62
0e36f-6s26													

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
123c3-ropc_xits_09_UNSAT	26	U	8.83/41.5	8.08/14.2	7.43/13.5	10.6/16.1	7.49/40.5	7.43/41.6	?	?	12.3	9.98	12.1
12535-sbdp4_138_sanitized	26	U	3880/8510	4090/6410	3190/5450	TO	3650/9540	3380/40600	?	?	?	?	3550
12820-RoundRobin_n16_d14	25	U	0.068/0.382	TO	TO	TO	0.266/0.517	TO	?	?	?	?	0.068
12916-hantzsche_wendt_unit_93	26	S	49.4	723	514	324	353	0.116P	110	135	158	700	748
12ac1-exam_75_65_sanitized	26	U	0.623/32.7	TO	TO	TO	0.317/22.2	TO	?	?	?	?	0.517
12b4a-Folkman-175-7416734_sanitized	24	S	138	39.1	132	129	76.3	149	?	?	?	?	152
12cd0-4pipe	26	U	4.12/22.4	3.02/6.84	3.62/10	2.87/7.19	2.47/18	3.82/27.5	22.5	4.12	10.2	4.52	3.07
1397d-adv_gc_n300_k10_UNSAT_s42_sanitized	26	U	3.87/12.6	2.62/4.74	2.47/4.99	7.13/14.2	1.573/7.1	4.37/28.1	63.9	0.867	2.52	1.97	3.57
13ae2-x9-10027.sat_sanitized	24	S	31.3	26.4	24.5	68.9	29.3	25.2	262	1.307	54.8	28.8	6.22
14273-shuffling-2-s242449-of-bench-sat04-727.used-as.sat04-753	24	S	42.6	4.37	17.5	23.3	27.9	4.77	3040	4.72	35.7	26.7	?
147ae-1x1x13_1x3x6_pair_0_0_0_2_2_1_sanitized	26	S	2100	485	900	340	283	2950	1480	2790	4200	483	3690
14e4c-x9-12035.sat_sanitized	24	S	267	180	179	58.2	194	183	442	36.8	65.8	190	0.567
14ea5-af-synthesis_stb_50_40_9_unsat	24	U	962/3160	792/1470	865/1590	1270/2210	974/3380	878/14500	1550	1110	950	899	938
1507d-x9-12014.sat_sanitized	24	S	182	53.6	13	58.5	59.4	13.4	2860	128	549	58.4	189
15456-hwmc20miters-iso-rast-p06_sanitized	24	U	1.63/8.67	0.216/1.53	0.216/1.53	0.216/1.58	1.27/5.86	0.366/8.54	?	?	15.1	4040	1.17
15456-clique_n5_k10_sanitized	26	U	1.43/2.39	7.63/21.6	12.1/38	17.9/50.7	1.72/2.85	0.066/0.051	?	?	?	?	1.32
15489-mdp-28-14_unsat	24	U	1100/3110	761/1460	1170/2240	1570/3110	827/2780	1180/12200	2380	2340	775	791	1830
1576e-6s299b685_lter22	25	S	6.13	1.42	6.38	6.02	2.92	6.43	0.518	41.2	1.37	3.27	7.2
15e66-adv_gc_n200_k12_UNSAT_s42_sanitized	26	U	5.84/20.3	13.3/31.3	43.7/8.64	57.9/157	18.9/91.4	8.63/70	?	?	28.5	3.97	19.8
16564-ER_400_20_7_apx_2_DS-ST	25	S	381	294	388	82.7	2220	412	?	?	557	1830	306
166e1-mp1-squ_any_s09x07_c27_abix_UN	26	U	233/556	286/499	234/403	254/438	290/1000	245/3230	376	460	320	275	287
16801-schooltt-5-5-9-2-4-5-10-0.1-0.2-seed2_sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
16c27-rbsat-v760c43649gyes7	24	S	4.4	476	73	156	526	3.72	1120	114	454	4.43	4.42
16c54-two-trees-511v_sanitized	24	S	0.059	1990	TO	TO	2160	0.066	1470	1490	1990	0.018	0.018
16e99-aaal10-planning-ipc5-pathways-17-step20	25	U	87.8/355	65.9/215	63.9/231	96.1/375	62.1/235	66.6/860	295	147	203	81.6	111
16ca5-mdp-36-10_unsat	24/26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
16f8b-zeroir.i.3.29_sanitized	26	U	4.15/33.5	TO	TO	TO	0.27/33.3	TO	?	?	?	?	?
16ff4-sort-mitem170	26	U	166/422	146/286	140/259	210/387	17/458	141/1390	324	256	247	155	167
1707c-xorshift_r16_119_sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
170a7-af-synthesis_stb_50_140_0_unsat	24	U	481/1560	449/847	385/883	653/1150	546/1900	417/5330	547	525	640	479	397
170b1-circuit_48in64out_with_800gates_4in4out_dist128_seed4_sanitized	24	S	10.3	10.9	10.1	8.28	9.49	91.8	10.3	14.1	9.7	14.2	?
1753a-GP_190_225_30	25	S	6.33	3.22	3.22	3.17	24	2.67P	2600	477	4.68	3.42	5.57
180e6-baseballcover1with22_and2positions	26	S	19.4	264	206	207	273	204	2130	895	?	230	30.9
1890f-homer11_shuffled	25/26	U	0.095/0.123	44.1/103	43.7/98.4	56.4/128	0.034/0.087	0.116/0.018	207	61.4	41.5	0.02	0.018
18c0e-cliquecolouring_n31_k5_c4_sanitized	24	U	0.121/0.479	TO	TO	TO	0.067/0.253	TO	?	?	?	?	0.117
18d41-spg_300_300	25	U	195/153	135/79.8	77.7/171	139/204	139/21.7	179/414	158	?	?	326	108
19585-x9-11077.sat_sanitized	25	S	192	159	37.1	66.5	177	37.6	462	115	65.1	173	9.63
19951-oddball_80_5_tto_zp_normalised	25	S	13.8	TO	TO	231	TO	2.52P	?	?	?	?	2270
19e2c-x9-09024.sat_sanitized	24	U	21.1/47.1	17.5/22.2	17.6/23.2	25.7/31.6	19.5/49.3	18/108	54.8	45.6	18	19.5	27.6
1a1c9-af-synthesis_stb_50_200_4_unsat	24	U	271/962	228/613	193/537	345/873	320/1220	195/1840	244	256	372	226	213
1a20d-hwmc20miters-iso-rast-p11_sanitized	24	U	1.62/8.61	0.216/1.63	0.216/1.58	0.216/1.53	1.47/6.17	0.366/11.2	?	?	15.7	3960	1.22
1a3be-le450_25b.col.25	26	U	2.02/12.2	TO	TO	TO	0.267/11.7	TO	?	?	?	?	0.567
1a929-BubbleVsPancakeSort_8_4	25	U	469/1640	407/870	396/832	565/1160	407/1410	1000/4120	3140	1540	1120	414	602
1a936-arles_thres10_p10_r8185	25	U	0.02/0.254	0.009/0.1	0.010/0.1	0.010/0.099	0.035/0.113	0.116/0.052	0.019	0.018	0.018	0.018	0.02
1a951-linked_list_swap_contents_safety_unwind50	25	U	59.5/2990	26.4/527	66.3/11100	56.4/9930	39.9/1030	9.58/157	57.4	361	86.8	57.3	71
1afa2-worker_550_550_550_0.3	24	S	19.4	11.5	11.7	11.8	79.9	17.8	?	?	205	462	19.1
1bf04-af-synthesis_stb_50_140_1_unsat	24	U	952/3070	1010/1860	971/1810	1220/2160	991/3320	855/10260	1620	1150	1190	887	858
1b3c7-rook-56-0-0	24/25	U	1900/4070	1240/1840	1480/2280	1320/2060	1250/1220	0.567/0.466*	2630	2370	2880	1250	1700
1b84d-ramsey_4_4_18_normalised	25	U	0.167/7.87	TO	TO	TO	1080/320	TO	?	?	?	?	?
1b880-oddball_52_5_tto_zp_normalised	25	S	6.08	TO	4780	41	1250	1.37P	3530	?	?	4550	5.02
1b9c3-cabp-M-bcsstk06.mtx.rnd-k33_sanitized	26	S	432	424	518	132	331	517	21.7	128	93.6	433	443
1b919-bp4_BC012_AM_FPBEQ_ZR_normalised	25	S	2050	3230	1690	451	2620	1670	?	?	3100	4640	822
1bb9f-goldd-heqc-k2mul	26	U	14/60.1	9.43/28.6	11.43/2.2	12.2/31	12.7/80.4	11.7/97.4	657	348	264	13	17.4
1c0fe-SDP_120_17_1675_sanitized	26	S	48.8	217	39.6	388	110	42.6	?	?	?	153	1410
1c21a-9_normalised	25	S	67.9	31.4	31.5	30.4	77.8	13.5P	?	?	103	36	33.1
1c308-oddball_26_5_ttf_normalised	25	U	776/2210	484/617	824/914	301/491	693/1470	848/13400	155	101	184	756	856
1d188-SC25_Timetable_C_392_E_45_CL_25_D_7_T_50_normalised	25	S	53.4	10.5	19.1	14.3	14.8	133	18.1	26.6	12.4	16.1	43
1d90b-multiplier_16bits_miter_22	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
1e0d5-QG7-genys-icl006.sat05-3132.res shuffled-07	25	U	256/585	161/284	182/325	267/476	107/316	184/2010	527	495	224	197	360
1e0da-linked_list_swap_contents_safety_unwind63	24	U	114/1880	51.4/1100	48.5/14300	48.8/14800	85.7/2300	16.5/315	158	558	104	89.1	111
1e3c3-app-sbox5-cut3-symmbreak	24	S	1530	TO	TO	TO	TO	TO	?	?	?	?	?
1e742-mdp-32-10_unsat	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
1e874-crusti_g2io_175_0_2_511_32_normalised	25	S	153	6.58	284	291	14	296	2.87	212	36.5	14.5	164
1ea68-REGRandom-K4-L1-Seed30_sanitized	24	U	155/3720	2.57/300	2.67/338	2.77/355	14.9/4460	3.47/184	?	?	236	444	5.52
1eea3-par32-4_shuffled	26	S	TO	TO	390	TO	TO	403	?	?	?	?	964
1f44d-REGRandom-K3-L3-Seed25_sanitized	24	U	9.13/143	0.917/151	0.917/146	0.867/145	10.6/2110	1.22/10.8	?	?	53.1	16.8	8.43
1f81f-ACG-20-10p1	26	S	149	209	227	166	152	824	567	107	726	207	194
1fac9-adv_gc_n100_k10_UNSAT_s42_sanitized	26	U	1.02/2.75	0.566/0.933	0.817/0.933	0.617/0.983	0.967/12.3	0.867/4.69	66.5	0.667	0.969	1.07	0.969
1fb64-chnl-040x041_sanitized	26	U	0.468/52.8	TO	TO	TO	0.116/50.4	TO	?	?	?	?	0.217
20273-chnl-050x051_sanitized	26	U	1.17/198	TO	TO	TO	0.166/192	TO	?	?	?	?	0.669
207b2-pcmach-scheduling-m37-28831-324346-SAT_sanitized	24	S	TO	2570	TO	TO	TO	TO	?	?	?	?	?
207eb-hypp_cec_multi_2	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
20ae1-bwoadm_28_5_sanitized	26	U	2720/6580	3130/5380	2740/4860	TO	3370/9340	2760/32200	?	?	?	?	3190
210fe-32_325_sanitized	24	S	90.9	61.4	122	124	98.5	141	?	?	?	96.5	67.2
21151-rook-42-0-1	24	U	860/2150	623/329	781/556	743/436	644/371	0.266/0.216*	1000	413	581	839	720
211b0-sbdp4_65_sanitized	26	U	3770/11400	3440/7140	3770/7980	TO	3690/14000	3760/28600*	?	?	?	4440	3350
21415-clcql_50_6_5_normalised	25	U	3.52/5.5	TO	TO	TO	0.166/0.799	0.166/3.39	?	?	?	?	3.42
218f0-sc_9630_26_0.3_0.1_17.af_76_sanitized	26	S	2.57	0.567	0.567	0.567	3.48	1.17	5.92	3.32	1.37	2.42	2.52
21b17-shuffling-1-s1769330284-of-bench-sat04-422.used-as.sat04-561	25	U	88.1/175	71.3/67.6	65.1/75.6	104/101	75.3/113	78.9/574	162	644	157	80.7	90.8
2241a-crusti_g2io_225_0.1_31_25_normalised	25	S	31.75	29.2	19.9	46.1	32.8	40.6	0.768	22.8	147	39.8	15
22465-as-p81-i27-o6-rot-568_sanitized	26	S	TO	4410	TO	TO	TO	TO	?	?	?	?	?
22b4c-j3037_10_rggt_b	26	S	12.4	19.5	53.6	30.5	3.12	56	77.2	6.02	27.5	31.7	7.78
22c8d-xor_op_n40_d3	25/26	U	143/338	1280/1370	1340/1320	2020/1920	101/485	2.12/1.63	1800	606	2140	1610	146
22df1-x9-10070.sat_sanitized	25	S	24.7	20.7	44.1	29	22.8	44.8	38.1	77.7	38	22.5	31.7
231f3-xorshift_r14_42_sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
23dae-mdp-32-14_unsat	24/26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
24075-UR-15-10p0	25	U	173/1810	141/1980	120/3300	121/3050	115/1850	553/24100	197	170	161	197	179
24123-arles_thres20_p10_r7340	25	U	0.068/0.141	0.010/0.084	0.009/0.084	0.009/0.083	0.066/0.094	0.034/0.036	0.018	0.2			

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
2c3c2-mchess_17	24	U	13.9/102	2070/3730	2160/3950	3200/4940	TO	0.116/0.026	4330	?	?	2990	0.019
2c449-1x1x14_1x2x9_pair_4_3_0_2_3_1.sanitized	26	S	1410	TO	3700	514	3330	4410	?	?	?	4140	?
2c458-oddball_70_5_tto_zp.normalised	26	S	10.7	4090	4120	115	4440	2.07P	112	?	?	2170	8.63
2c4eb-oski15a01b45s_opt	25	U	679/3430	773/2790	555/2250	821/2270	390/3530	481/6850	4730	4210	?	803	546
2ca47-st_540_25_12_12_9.af.sanitized	24	S	240	2330	2880	TO	TO	3050	2960	?	?	238	?
2d0e0-170223547	25	S	1.72	1.47	1.42	60.1	1.57	2.27	?	?	?	19	1.57
2d261-ntil-90d-44.sanitized	26	S	TO	TO	TO	TO	TO	TO	?	?	?	4990	?
2d6d2-fixedbandwidth-eq-31_shuffled	24	U	1240/3090	675/791	753/671	1220/1260	907/2510	790/6950	?	808	2540	661	1060
2d89f-2.sanitized	26	S	0.217	TO	TO	TO	TO	0.066	?	?	?	0.217	0.518
2d44b-rhnp_p8_r160.sanitized	24	U	23.4/405	TO	TO	TO	0.317/672	0.817/148	?	?	?	0.567	0.52
2d9b-case11.normalised	25	S	362	714	800	266	768	1010	151	934	301	734	164
2e058-blocks-blocks-36-0.150-NOTKNOWN	25	S	81.4	67.1	91	78.2	75.6	237	7.88	4000	374	80.2	78.3
2eb03-ER_500_30_3.apx_1_DC-ST	24	U	2047/375	180/378	171/360	177/381	174/923	177/2210	190	163	241	186	210
2fd8-simon-r20-0.sanitized	24/26	S	172	0.009	0.009	0.009	0.067	133	?	0.018	0.018	0.066	164
30056-DLTM_twitter634_77_13	26	U	3870/5610	3080/3720	3430/3810	TO	3290/7110	3910/45000	?	3790	3560	3060	?
30348-circuit_48n64out_with_700gates_4in4out_dist128_seed1.sanitized	24	S	2.37	2.92	1.17	2.52	6.22	4.47	11.5	8.03	5.94	4.17	5.24
305ad-asconhashv12_opt64_H9_M2-MIC4kfhia_m0_6_U2.c	24	U	247/637	264/860	285/1070	281/1010	253/846	259/4260	?	?	248	226	293
306a2-lattice-25.24.sanitized	26	U	3.77/17.8	TO	TO	TO	10.3/188	TO	?	?	?	0.969	1.12
30c6f-6.sanitized	26	S	0.217	TO	TO	TO	TO	0.066	?	?	?	0.217	0.22
30e8b-aeq_equiv_encry_3_rounds.debugged	26	U	407/6740	364/1540	279/17900	360/17700	268/17700	627	386	523	393	305	305
30f0d-Folkman-190-66337703.sanitized	24	S	344	128	141	237	202	149	?	?	1080	125	224
30f64-lockchart-group1-L200-K289-p8d4j1.normalised	25	S	1290	2140	1970	389	2400	77.8P	?	?	?	1390	1240
31291-simon-r24-1.sanitized	24	S	194	0.01	0.01	0.01	0.067	141	?	0.018	0.018	0.066	189
3184e-mdp-28-10-unsat	24	U	562/1690	499/1090	515/1080	1250/2600	570/2160	544/6000	913	1060	796	490	1520
3197b-b20_1	25	U	6.43/12.5	4.92/4.04	5.18/5.44	4.52/4.79	5.23/7.06	5.37/18.8	21.9	22.8	9.73	6.42	6.27
31bb6-hwmccl17miters-xits-iso-oski15a08b08s.sanitized	24	U	4.17/59.2	3.87/34.4	4.02/43.1	4.72/1580	4.34/49.6	TO	?	?	210	7.63	3.62
31e78-pcmax-scheduling-m40-26287-324155-SAT.sanitized	24	U	TO	TO	TO	TO	TO	TO	?	?	?	2660	?
31e84-sqrt-mitern171	25	U	76.2/202	68.8/130	66.3/127	79.6/144	77.2/214	70.2/276	111	108	120	70.8	80
31fa0-cabp-P-can_445.mtx.rnd-k96.sanitized	26	S	314	336	1360	121	156	334	?	?	247	299	294
32173-16_16_booth_wallace_mapped_and_default_origin_bit28	25	U	782/2170	794/1550	735/1460	2250/3170	884/2380	776/9200	3130	2830	1050	832	2680
324b2-SC21_Timetable_C_512_E_66_C1_34_S_35	26	?	?	?	?	?	?	?	?	?	?	?	?
32585-pcmax-scheduling-m26-6398-62377-UNSAT.sanitized	24	U	133/642	122/382	107/335	135/410	72/424	108/1150	752	295	103	50.7	93.2
32e34-lightsout_sat_24_unbounded_totalizer_corner_0_unsat.sanitized	26	U	0.719/7.36	TO	4540/4630	TO	TO	4300/44900	?	?	?	4750	?
3344a-HCP-446-105	24/25	S	913	890	998	879	910	0.266P	2910	1580	1370	855	835
3367b-shuffling-2-s158913847-of-bench-sat04-306.used-as.sat04-610	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
337ee-SocialGolfers-6-6-7-cp_c18	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
33c8a-SDP_96_14_3612.sanitized	26	S	92.3	10.6	26.8	380	39.9	27.1	2840	322	61	39	4.12
34360-Large-result_b23.af_303.sanitized	26	S	2.59	0.717	0.667	0.717	580	1.12	1.02	2.37	1.97	1.47	2.42
34fa0-F9idw	24	U	237/708	170/284	226/291	236/247	199/267	221/1260	352	673	391	236	237
35748-oski15a01b39s_opt	25	U	455/3580	390/2770	373/2180	395/2080	359/3730	714/7060	3880	4070	?	467	353
35789-iseitingridx200_shuffled	24	U	0.035/0.488	TO	TO	TO	14.1/8.62	0.066/0.382	?	?	?	17.5	0.018
35938-Break_18_32.xml	24	S	TO	TO	3490	1040	3770	TO	?	?	?	3210	?
3598c-SGI_30_70_22_70_7-dir.shuffled-as.sat03-154	26	U	3970/18500	3400/11900	3540/12000	3890/3220	3350/1160	3580/1.42	?	?	?	4010	3770
35b90-bp4_TCO_CS0_ZR.normalised	25	S	TO	3310	782	1650	699	863	?	?	?	3700	4300
35e0b-count_p2_M33.sanitized	26	U	0.518/1.67	TO	TO	TO	0.367/1.25	TO	?	?	?	0.517	0.517
35ec9-MVRoundRobin_n14_d10_v2	25	U	2.02/13.6	TO	TO	TO	0.918/20.6	TO	?	?	?	1.92	2.02
36191-aries_thres20_p10_r7475	26	U	0.067/0.14	0.01/0.084	0.01/0.084	0.01/0.084	0.034/0.094	0.066/0.036	0.018	0.217	0.018	0.034	0.034
362d1-Carry_Bits_Fast_19.cnf	25	U	3.27	1.62	0.516	0.817	1.42	0.066	172	2.97	2.12	0.768	0.567
365a0-Medium-result_h10.af_252.sanitized	26	S	1.62	0.417	0.417	39.5	0.968	0.618	1.42	1.17	1.42	1.52	?
36ba4-comm-ue-csp-sat-n1200-d-0.02+0.05595518.shuffled-as.sat05-531	26	S	3610	TO	TO	553	TO	TO	?	?	?	?	1280
36dbc-iseitin_grid_n11_m20.sanitized	24	U	0.02/0.113	TO	TO	TO	0.166/0.234	0.116/0.068	341	?	?	0.367	0.018
37202-13045_4_rggt_bm1	26	U	271/683	259/427	229/406	312/504	281/809	235/2460	1380	903	328	282	265
3732b-as-p123-139-o0-rot-II-4680-i.sanitized	26	S	2810	2010	2250	1880	2470	1540	2000	1180	3290	2150	2040
37463-6s299b685_Iter30	26	S	62.8	46.4	55.7	63.4	97.6	60.9	34.5	339	787	58.5	59.2
37a11-mdp-28-16-unsat	24	U	978/2660	880/1830	920/1830	1180/2010	967/3520	929/9830	2670	1670	972	1020	926
37c88-act_enc1_opt2_20_12	26	U	1190/1430	862/1110	889/1110	1090/1390	968/2500	869/3890	943	959	1710	1130	1370
37ca1-circuit_32in32out_with_350gates_6in6out_dist64_seed1.sanitized	26	U	0.719	44.4	37.9	45.6	51.7	277	175	116	801	62.9	0.719
37fa6-SDP_106_15_2616.sanitized	26	U	969	50.7	199	436	329	196	4440	?	125	57	55.3
38504-ec_mult_CvD_11x11.sanitized	24	U	451/1660	447/1140	390/977	927/2410	491/1950	395/1410	1920	2280	1060	424	492
389bb-BubbleVsPancakeSort_7_6	26	U	240/770	217/398	182/322	301/458	230/670	787/1770	1610	605	492	224	298
39277-crafted_n12_d6_c4_num23	24	U	298/28800	179/33000	150/18200	228/30500	191/28800	165/5780	3890	3900	349	262	237
39582-lockchart-group2-md0.3-L19-K38-P8D4J1_1.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
396fd-circuit_32in32out_with_80gates_7in7out_dist128_seed1.sanitized	24	S	0.919	73.6	75.2	75.8	81.4	23.8	2670	692	138	134	0.918
3988a-Break_08_24.xml	24	S	0.218	0.116	0.066	0.066	0.316	0.116	0.035	0.367	0.217	0.217	0.217
39fba-SGI_30_80_26_70_4-log.shuffled-as.sat03-208	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
3a1f1-MVRoundRobin_n16_d10_v3	25	U	6.83/95.3	TO	TO	TO	2.57/600	TO	?	?	?	7.28	6.28
3a75a-x9-11093.sat.sanitized	25	S	55.7	47.4	24.5	23.9	52.5	25.1	2.12	181	36.5	51.3	26
3a840-x2_64.shuffled-as.sat03-1603.cnf.mis-125.debugged	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
3a9df-ntil-90d-40.sanitized	26	S	101	TO	223	2280	TO	TO	?	?	?	?	104
3b0ee-as-p153-157-o0-rot-rfl-II-68-i.sanitized	26	S	992	4030	2320	3430	TO	1240	1360	2630	3140	981	?
3b560-cabp-X-can_715.mtx.rnd-k108.sanitized	26	S	452	480	442	554	592	1070	?	?	764	3940	478
3b59c-g2_T93.2.1	24	U	518/3990	429/2870	412/3990	509/4290	368/462	460/7640	385	1140	664	497	507
3b5d7-nla-digbench-scaling_dijkstra-u_valuebound1_transition	25	U	1.02/1.24	0.517/3.14	0.517/3.14	0.517/3.14	1.02/3.31	0.917/24.6	?	?	23.3	?	0.968
3b608-Medium-result_b9.af_268.sanitized	26	S	1.47	0.316	0.317	0.317	34.6	0.818	0.468	1.02	0.818	1.37	1.42
3bf8b-summlle_X4044_steps7_11-2-2-4-4-8-25-100	24	S	8.38	6.02	5.67	8.03	12.2	0.317	49.6	26.1	5.68	11	8.38
3bfa9-case6.normalised	25	S	1530	850	380	334	899	386	790	?	94.3	879	642
3c0db-rook-52-0-1	25	U	2460/4520	2860/1680	2690/1820	1720/1610	2760/1110	0.467/0.366	?	?	?	1480	2370
3c129-oski15a01b09s_opt	25	U	420/3560	444/2790	324/2080	347/2210	344/3680	518/8640	4620	4590	?	409	374
3c4ac-homer18.shuffled	26	U	0.036/0.162	320/703	232/627	258/601	0.034/0.146	0.066/0.02	2040	1240	208	0.018	0.018
3c51a-fastlec_cep_14_T27_58.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
3c6e1-x9-10031.sat.sanitized	24	U	185/385	156/209	149/197	300/421	168/501	152/964	1060	774	177	168	331
3cda1-lockchart-group2-rnd0.3-L19-K38-P8D4J1_3	25	S	1250	4680	TO	986	TO	TO	?	?	?	314	320
3e449-newpol23-4	26	U	51.6/33.5	228/39.8	82.8/18.4	82.4/39.4	70.2/19.4	98.1/949	170	478	853	227	41.4
3e7d8-bdbmwm_28_94.sanitized	26	U	2410/6060	2240/3680	2250/3630	TO	2420/6590	2310/23700	?	?	4010	2320	?
3e912-grain-53-00-0s0-seed-125-4-init-35	26	U	192/500	171/298	180/296	289/520	179/554	182/1590	259	432	285	179	318
3ec81-gm16spctrc	25	U	797/1490	709/604	560/463	490/324	568/717	39.9/419	146	78.4	191	708	599
3ed56-rbsat-v945c61409gyes9-sc2009	24/26	S	79.5	63.5	16.8	11	66.7	68.4	136	54.9	162		

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
47e1a-noL-11-8.sanitized	24	S	TO	422	98.9	499	2240	0.066	?	?	1510	41.3	43.5
47f58-b17	25	U	39.5/59.7	24.5/25.7	32.7/28.7	33.2/24.5	25.9/50.5	33.3/195	33.3	52.3	34.9	31.1	40.6
48229-myciel6-cn-used-as.sat04-319	25	U	360/1090	484/718	517/797	910/1390	610/2050	548/4180	2500	1220	703	535	368
48432-schooltt-5-3-8-2-4-5-10-0.1-0.2-seed2.sanitized	26	U	126/535	51.6/88.4	42.9/68.6	43.6/64.4	113/438	45/393	?	?	179	147	150
49644-cabp-X-can_715.mtx.rnd.k116.sanitized	26	S	TO	TO	TO	TO	1860	4720	TO	?	?	4500	?
49892-asconhshv12_opt64_H7_M2-K1zAs8_m0_3_U5.c	26	U	117/236	91.3/207	107/210	204/195	92.8/242	407/1550	485	345	108	104	123
4a828-1-TC-256-K-68.sanitized	24	U	99.7/332	105/257	103/255	81.2/242	93.5/534	102/889	86.6	69.3	119	88.9	110
4aab4-clccl_25_8_7.normalised	24	U	0.117/0.751	27.7	TO	TO	0.066/0.365	0.116/1.63	?	?	?	0.116	0.117
4ae68-x9-10032.sat.sanitized	26	S	68.2	57.1	64	46	63.2	65.9	491	47.2	0.669	61.5	21.7
4ba2c-at-least-two-vmpc_28	25	S	1320	1140	1450	916	1450	1270	302	373	1730	1430	1640
4be4a-ER_400_20_4.apx_1_DC-AD	24	U	307/1130	211/496	213/492	293/638	261/1410	220/2920	346	410	331	241	311
4c037-lockchart-group2-rnd0.3-L18-K36-P8D4J1	25	S	446	1160	1990	56.5	546	74.2P	?	?	110	884	371
4c300-fsf-300-354-2-2-3-2.9.opt	25	S	104	83.2	86.4	88	93.8	8.43	?	?	750	892	73.6
4c54e-lec_mult_CvK_12x11.sanitized	24	U	3070/7850	2630/6000	3000/6530	TO	2850/9530	2950/35700	?	?	?	3090	?
4c7e0-Ptn-7824-b19	25	S	32.3	1070	58.2	30.2	1170	0.867	1350	13.3	2280	1090	39.8
4cf35-odball_26_4_ttf.normalised	25	U	1120/3230	814/1060	1120/1160	915/1140	777/1760	1140/18300	82.3	44.7	236	890	683
4d6bd-vdw_4_8_n145.sanitized	26	S	1690	3530	3840	TO	1570	3730	?	?	2810	3490	1600
4d6e1-GP_100_951_33	25	S	14	7.83	7.68	7.63	23.2	6.08P	?	?	10.6	10.9	7.98
4d7f6-stb_418_125.apx_1_DC-ST	24	U	341/957	297/492	290/470	506/823	315/1190	288/2600	467	532	327	309	585
4d84e-4.sanitized	26	S	0.218	TO	TO	TO	TO	0.066	?	?	?	0.217	0.217
4de5b-as-p141-139-o0-rot-rfl-680-i.sanitized	26	S	2110	1990	2290	TO	1790	3120	2310	2290	1940	3050	4910
4e0de-9.xz.sanitized	26	S	14.5	8.98	8.83	8.68	24.5	8.63	8.33	22.6	10.6	9.13	13.9
4e319-case1.normalised	25	S	106	589	395	461	683	397	522	903	521	106	606
4e366-Dodecahedron-k7	24	U	TO	TO	TO	TO	TO	TO	?	?	?	?	?
4e64f-satcoin-genesis-UNSAT-7200	26	U	478/921	624/1020	805/1030	TO	304/719	0.467/3720	?	?	436	501	551
4f09e-BubbleVsPancakeSort_6_7	26	U	67/229	36.8/66.3	43.5/85.5	62.7/108	48.5/138	107/646	109	89.6	66.2	39.4	77
4f2d6-mdp-32-11-unsat	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
4f4d2-xor_opt_n38_d3	24	U	122/320	699/717	690/715	954/774	77.9/318	1.821.38	?	?	1620	2110	138
50019-noL-11-14.sanitized	25	U	26.5	27.7	35.9	38.8	21.5	0.033	?	?	78.5	31.3	15.7
50519-arlex_thres10_p10_r8142	25	U	0.02/0.256	0.009/0.099	0.009/0.099	0.009/0.099	0.034/0.093	0.034/0.051	0.018	0.018	0.019	0.019	0.019
510df-51-136961	26	S	943	151	176	41.9	173	59.6	?	?	185	170	1240
511a2-circuit_32in32out_with_500gates_6in6out_dist64_seed1.sanitized	26	S	1.02	110	90.1	76.7	119	3.62	1200	191	175	147	0.968
511f1-14.sanitized	24	S	0.218	TO	TO	TO	TO	0.066	?	?	?	0.217	0.216
51482-hob12s09-opt	25	U	49.5/99.9	43.1/53.8	34.3/46.1	41.5/50.4	45.8/92.3	46.9/406	236	172	48.8	47.2	45
51551-SDP_134_18_4334.sanitized	26	S	TO	TO	TO	TO	TO	TO	?	?	?	?	?
5220b-coreMQTT:MQTT_ProcessLoop.dimacs	26	U	160/1310	104/631	55.5/5610	63.3/5880	122/5440	96.1/13400*	68.3	1230	238	150	154
5246b-b15	25	U	4.84/12.9	2.82/5.09	2.82/4.49	3.17/4.79	2.72/7.62	2.87/19.6	2.87	5.92	3.92	3.72	3.52
526ca-ramsey_3_7_23.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
52aaf-noL-11-10.sanitized	24	S	46.2	180	47.2	36.4	98.9	0.034	?	?	?	350	?
5314f-oski15a01b15s_opt	25/26	U	516/3790	341/2860	700/2490	380/2280	488/3220	666/10800	4320	3070	?	649	494
53301-baseballcover12with25_and5positions	25	U	3770/7240	3750/4000*	3090/4000*	2990/4000*	3920/2880*	3180/1200*	?	?	?	2460	2630
539a1-multiplier_15bits_miter_23	25	U	1320/3160	1130/2090	1300/2480	3120/5360	1250/3490	1250/14000	?	?	3520	1580	1140
53c21-maximum_constrained_partition_14_bits_n200	25	S	5.02	26.7	48.4	21.6	9.62	0.517	16.4	15	22.8	47.6	12.7
53d23-puzzle32_sat	25	S	835	109	762	12	2360	0.116	383	65.2	583	128	4250
53e34-RoundRobin_n17_d15	25	U	0.117/0.459	TO	TO	TO	0.617/0.751	TO	?	?	?	0.117	0.117
53e8a-shuffling-2-s504782760-of-bench-sat04-312.used-as.sat04-604	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
54110-lockcrest-and-9	24	U	116/2470	91.1/2570	103/2860	113/3100	98.1/2230	113/2390	277	338	190	160	105
5432e-pcmxax-scheduling-m24-24102-255206-SAT.sanitized	26	S	TO	TO	TO	TO	1600	TO	?	?	1140	?	?
543e6-qwh.60.1728.shuffled-as.sat03-1659	24/26	S	55.5	34.4	36.3	26.7	36.7	125	37.7	66.3	27.1	41.6	64.8
546a7-schooltt-5-3-8-2-4-5-10-0.1-0.2-seed1.sanitized	26	U	213/822	99/117	89/101	102/76.1	155/626	153/1300	?	?	275	184	158
546f8-intel047_Iter78	25	S	11.8	7.18	7.83	8.18	30	8.58	32.2	60	248	10.7	14
54c2d-x9-10038.sat.sanitized	24	S	72.1	61.2	92.5	52.4	67.9	93.5	109	130	69.2	66.3	71.1
54d49-hwmccl17miters-xits-iso-6s163.sanitized	25	U	0.518/33.2	0.317/3.02*	0.316/3.07*	0.316/3.07*	11.1/14.46*	0.867/20.2	0.668	0.668	0.719	0.418	0.468
55199-WS_500_16_90_70.apx_2_DC-AD	24	U	290/815	237/376	207/326	435/676	302/966	212/1750	409	363	299	281	544
553b0-ntl-90d-46.sanitized	26	S	TO	TO	TO	TO	TO	TO	?	?	4770	?	?
5553f-post-cbmc-aes-ce-r2	26	U	112/6520	83/5370	65.2/5080	85.4/5240	91/7810	69.3/5190	215	100	105	112	103
556cf-tseitin_d3_n174.sanitized	24	U	0.018/0.043	TO	TO	TO	0.066/0.034	?	?	?	?	0.116	0.019
557d7-mp1-Nb7T46	25	S	10.7	9.08	249	94.5	10.9	7.88	?	?	28	293	21.6
5690b-170058440	26	S	682	580	581	1140	628	10.1	?	?	1710	608	264
56913-grs-32-64	25	U	112/513	73.5/234	77/247	81.7/213	89.9/425	75.5/1240	2860	577	108	83.4	111
573cc-linked_list_swap_contents_safety_unwind80	24	U	118/3050	68.1/2030	97.6/26300	280/4000*	79.8/6080	28.6/525	392	833	85.5	157	111
576eb-j3037_9_rggt_b	24	S	62.7	144	189	21	204	199	17.2	136	123	167	203
57b47-VanDerWaerden_pd_2-3-22_462	25	S	12.2	852	1370	897	900	56P	798	1030	2140	842	886
57f4e-x9-09047.sat.sanitized	24	U	19.3/45.2	22.2/28.5	16.5/21.9	26.5/29.2	24.4/63.5	16.9/97.5	102	80	21.2	24.3	34.4
57f6b-3.xz.sanitized	26	S	74.8	4.67	4.62	4.62	26.7	4.92	4.22	11.8	5.62	4.17	7.53
5831f-s38584	25	U	2.72/5.33	1.77/2.19	1.37/1.58	1.42/1.84	2.06/5.1	1.42/6.09	4.17	7.97	4.42	3.07	2.62
5865f-x9-11088.sat.sanitized	24	S	324	279	144	87.1	301	147	1240	1680	279	294	103
58eaf-count_p2_M41.sanitized	26	U	1.32/4.99	TO	TO	TO	0.967/3.8	TO	?	?	1.32	1.32	1.32
59116-oski15a01b06s_opt	25	U	913/3930	278/2790	378/2280	387/2200	326/3490	737/10700	4830	4720	?	588	455
5952b-ctl_4291_567_9_unsat	24	U	362/892	391/763	315/575	471/847	396/1260	359/6600	216	298	338	406	514
59825-hcp_CP18_18	24/25	S	407	1090	1940	150	474	1.12P	898	?	52.9	1500	884
59858-T92.2.0	26	U	421/2590	264/2620	366/5720	409/5900	249/739	371/6190	259	763	330	411	417
599c7-jgraledelevy_2200.9086.08.40.149-sr2015	25	S	189	135	135	3.1	138	15.5	80.8	15.5	38.1	13.8	13.8
5a068-ITC2021_Middle_1.xml	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
5ab30-g2-hwmccl5deep-6s161-k17	24	U	534/1310	485/721	562/849	562/686	600/1250	606/6830	?	?	4430	628	525
5ab5a-sbdp4_136_2.sanitized	26	U	3170/7210	2890/4850	2970/5110	TO	2900/9210	3350/35600	?	?	?	3710	?
5aed2-oisc-subrv-and-nested-12	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
5b0f1-lec_mult_KvW_11x10.sanitized	24	U	451/1280	385/853	372/823	914/1970	424/1340	375/3660	1980	2770	740	427	832
5b5df-pcmxax-scheduling-m15-2352-13561-SAT.sanitized	24	S	TO	TO	TO	TO	TO	TO	?	?	1670	?	2250
5babc-8.xz.sanitized	26	S	13.1	8.43	8.13	8.18	21.7	7.98	8.03	21.4	9.94	7.63	13
5c24c-c7552	25	U	0.669/0.551	0.416/0.332	0.316/0.332	0.266/0.282	0.517/0.502	0.316/0.834	0.367	0.718	0.618	0.668	0.668
5c29c-SGI_30_60_20_50_3-dir.shuffled-as.sat03-114	25	U	970/3740	723/2250	711/2230	702/2160	742/510	934/10500	1210	1260	1260	883	1080
5c478-oski15a01b20s_opt	25	U	615/3360	350/2890	634/2290	381/2270	740/3700	700/9750	4480	3610	?	578	484
5c8f4-case7.normalised	25	S	108	150	42.9	186	165	44.1	371	40	121	164	45
5cae5-jkkk-one-one-11-38-sat	26	S	2230	80.1	TO	765	115	2850	2940	302	265	91	532
5cb1c-stb_531_83.apx_1_DC-ST	24	U	181/492	150/243	166/275	246/414	176/643	168/1570	286	266	216	162	271
5ce7d-56-126575	26	S	144	268	179	41.2	293	18.3	420	73.6	320	239	246
5ceb9-pj2009_k80	25	S	138	76.8	89.3	75.9	152	112					

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
663bb-SCPC-500-13	25	U	9.38/21	7.42/10.5	7.78/10.8	15.2/16.2	8.49/26.6	13.4/2.89	6.52	7.13	9.63	8.63	19.4
66734-lightsout_sat_23_unbounded_totalizer_corner_0_unsat.sanitized	26	U	0.617/5.9	224/135	286/168	568/460	274/376	282/2250	?	?	221	231	407
67133-snw_16_9_Encpre	26	S	TO	TO	2060	1160	4780	260P	?	?	2480	?	1990
674af-crusti_g2io_250_0_2_255_18.normalised	25	S	20.2	10	130	82.3	17	159	9.23	269	80	16.9	19
67934-55-140792	26	S	TO	TO	TO	916	TO	TO	?	4030	?	?	?
67aa1-jkkl-one-one-10-34-sat	25	S	17.4	233	17.6	39	2.52	19.3	7.78	106	67.5	243	7.48
67b48-stb_588_138.apx_1_DC-AD	24	U	383/968	293/552	343/480	788/1170	418/1210	352/3230	782	644	584	436	941
67c53-cliquecolouring_n15_k9_c8.sanitized	24	U	0.118/0.367	TO	TO	TO	0.066/0.07	TO	?	?	?	0.066	0.067
6832F-velev-pipe-sat-1_0-b7	25	S	14.3	104	190	26.1	159	150	11.7	?	145	94.8	82.4
68c33-arles_thres10_n10_r8180	25	U	0.02/0.257	0.01/0.1	0.009/0.099	0.01/0.099	0.034/0.093	0.066/0.07	0.019	0.019	0.01	0.018	0.018
68ef4-vdwb_k6_n350.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
695d8-x9-07092.sat.sanitized	26	?	0.118	0.116	0.116	0.066	0.116	0.116	0.667	1.02	0.568	0.117	0.117
695e2-rbsat-v1150c84314gyes1	24	S	270	TO	TO	TO	TO	220	?	?	?	274	274
6969c-asconhashv12_opt64_H11_M12-fCHjS2L0du5_m2_4.c	24	S	180	28.1	152	204	127	7.23P	?	36.8	128	29.5	76.7
697c9-x9-11094.sat.sanitized	24	S	10.5	8.83	270	139	9.67	275	358	418	313	9.73	105
69ad7-BubbleVsPancakeSort_9_4	25	U	4860/17600	4430/10800	4490/10200	TO	4280/16600	TO	?	?	?	3680	?
69e34-sokoban-p16.sas.ex.15-sc2016	24	U	728/18200	643/17800	434/30000	424/31100	688/28800*	1200/13600	?	?	?	716	778
69ed2-crusti_g2io_250_0_2_255_31.normalised	25	S	17.5	10.2	128	108	17	95.6	18.7	232	253	17.3	19.8
69f6d-Break_04_04.xml	25	S	0.068	0.018	0.018	0.017	0.066	0.068	0.01	0.018	0.034	0.034	0.035
6ad74-1ZC-1024-K-116.sanitized	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
6ad45-dobdm_14_1.sanitized	26	U	2870/8460	2220/4520	2670/5340	TO	2230/7730	2700/28300	?	?	3970	2110	?
6add7-constraints_16_0_5_1.sanitized	24/26	U	65.1/219	56.6/94.8	56.6/89.1	69.3/110	63.1/197	63.2/333	160	128	68.6	70.6	90.5
6c52a-1-TC-256-K-71.sanitized	24	U	89.1/334	60/210	58.6/209	48.9/210	55/457	59.8/429	34.6	30.5	48.5	75.2	72
6cb99-RoundRobin_n16_d13	25	U	0.067/0.364	1400/1150	582/585	942/743	0.166/0.467	592/6420	?	981	?	0.067	0.067
6cd46-sembuster_7500.af_74.sanitized	26	S	2.37	0.467	0.417	0.417	1.17	2.37	0.719	1.07	0.818	1.47	2.17
6cd95-sted1_0x24204-330	26	S	130	68	108	82.2	32.3	0.667	178	153	204	80.5	116
6cfd9-GreenTao_2-4-5_4231	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
6d35b-contest04-lksat-n1100-m7545-k4-l4-s310659001.sat05-524.res shuffled-07	25	U	182/390	145/161	153/179	242/289	143/334	284/867	452	394	195	158	310
6d815-Carry_Bits_Fast_12.cnf	26	S	1.97	0.366	0.516	1.22	4.73	0.066	125	0.868	6.46	2.82	3.12
6db4b-color-19-4.shuffled-as.sat05-444	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
6ddda-inith.i.2.30.sanitized	26	S	7.98/99.1	?	?	?	0.517/103	?	?	?	?	1.47	1.52
6el39-x9-06068.sat.sanitized	26	S	0.518	0.416	0.416	0.566	0.467	0.466	0.467	0.317	0.216	0.467	0.567
6e297-spg_200_301	26	U	333/241	439/62	257/190	31.7/83.5	440/38.4	260/800	13.3	216	164	564	166
6e8db-newpol32-4	26	U	319/375	366/60.7	447/91	302/59.5	274/100	688/7060	570	1510	48.2	315	315
6ebe9-17.normalised	25	S	136	65.6	77	66.9	E1	30.9P	?	152	?	61.8	141
6ef09-summle_X8638_steps7_I1-2-2-4-4-8-25-100	25	S	33.6	13.3	10.4	28.5	12.8	0.316	26.2	30.4	10.7	13.5	18.4
6f0e2-rook-56-1-1	24	U	637/1570	465/822	531/1030	381/1060	456/563	0.617/1.22*	1340	1140	939	549	558
6f1b1-as-p93-127-o6-rot-rl1-5682-i.sanitized	26	S	122	141	84.3	159	172	91.6	186	119	201	90.2	37.4
6f354-bp4_CSO_LP_FPBLE_ZR_YS.normalised	25	U	125/425	95.8/268	111/304	121/321	103/487	113/1220	1370	253	107	124	127
6f40b-harder-fphp-016-015.sat05-1230.res shuffled-07	25	U	0.036/0.174	TO	TO	TO	0.034/0.094	0.116/0.02	?	?	?	0.018	0.02
6f4df-SGL_30_60_19_60_6-dir.shuffled-as.sat03-112	24	U	988/4020	771/2530	756/2540	798/2580	812/4870	964/12100	948	1310	915	992	1180
6f6a2-RoundRobin_n18_d15	25	U	0.118/0.586	TO	TO	TO	0.517/1.45	TO	?	?	?	0.117	0.117
6f730-case13.normalised	25	S	22.6	8.78	9.48	30.6	9.08	77.9	6.02	57	51.3	8.38	?
6f7a0-circuit_64in64out_with_64gates_8in5out_dist256_seed1.sanitized	26	S	157	23.8	23.6	27.2	25.5	137	59.4	95.1	25.2	89.6	170
6fab3-stb_664_50.apx_2_DC-ST	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
6fc52-pcmax-scheduling-m24-17855-226744-SAT.sanitized	24	S	362	TO	2920	653	783	2520	?	?	?	?	?
6ff70-crafted_n10_d6_c4_num9	25	U	85.5/2370	82.3/34000	70.1/31300	68.9/27900	121/28800*	70.4/1650	586	1930	203	86	86
7083b-battleship-14-26-sat	24	S	16.5	7.08	7.03	50.4	7.12	0.116P	?	80.4	1.52	3.22	3.32
708fb-circuit_32in32out_with_100gates_7in7out_dist64_seed2.sanitized	24	S	592	52.2	48.1	52.8	53.4	3.32	11	225	93.3	85.1	0.918
70da0-x-epic-a19-p15.transition	25	U	1.32/1	0.216/1.98	0.266/2.04	1.37/17	0.266/2.04	0.467/8.84	?	10.1	1.07	0.668	1.22
70ef2-qvhl_50_1250.shuffled-as.sat03-1655	25	S	13.1	10.9	9.43	3.17	11.7	8.68	5.68	4.97	5.67	13.9	9.31
71849-exam_85_70.sanitized	26	U	0.769/49.6	TO	TO	TO	0.317/33.7	TO	?	?	?	0.869	0.819
71a3d-af-synthesis_stb_50_120_4_sat	24	S	113	123	166	34.8	130	44.2	45.6	151	57.9	140	102
71ec9-circuit_32in32out_with_70gates_7in7out_dist128_seed1.sanitized	26	S	0.819	171	109	262	166	2.57	?	1800	457	208	0.768
727eb-count_p2_M25.sanitized	26	U	0.168/0.485	TO	TO	TO	0.166/0.348	TO	?	?	?	0.166	0.167
72a03-frb80-14-1.used-as.sat04-879	25	S	3.32	2180	1130	2380	2260	3.27P	?	4190	2560	3.42	3.42
72c0d-Folkman-175-1251868.sanitized	25	S	50.8	54.3	38.3	18.5	75.6	41.4	?	168	61	49.5	?
72cec-bp4_LPL_FPBEQ_ZR.normalised	25	S	TO	1810	1730	872	2110	1730	?	3440	?	?	1720
73721-DSC125.9.25.sanitized	26	U	0.417/20.6	TO	TO	TO	0.167/19.3	TO	?	?	?	0.318	0.317
7429e-af-synthesis_stb_50_100_9_sat	24	S	24.5	58.3	2.57	31.7	163	2.77	56.1	160	130	136	101
74423-10.sanitized	26	S	0.217	TO	TO	TO	TO	0.066	?	?	?	0.216	0.217
74931-DivS_862_11.cnf.sanitized	26	S	30.3	17.9	18.8	17.9	19.8	4.22	48.4	114	32.4	32.4	30.5
74f14-pcmax-scheduling-m43-38782-385402-SAT.sanitized	24	U	TO	TO	TO	2860	TO	TO	?	?	?	?	?
75375-16.normalised	25	S	16.8	7.38	7.28	7.23	19.1	3.62P	277	19.8	8.63	9.98	14.2
753ca-oski15a01b01s_opt	25	U	399/3640	338/3050	520/2430	414/2400	482/3560	614/6830	?	3540	?	394	349
75429-mod2-rand3bip-sat-250-3.shuffled-as.sat05-2220	26	S	2.04	111	182	85	121	0.034	403	501	166	0.019	0.019
757d6-chnl-030x091.sanitized	26	U	1.03/38.7	TO	TO	TO	0.216/35.8	TO	?	?	?	0.568	0.519
75b23-ntl-90d-43.sanitized	26	S	1930	TO	TO	TO	TO	TO	?	?	?	?	1870
75cc1-SCPC-500-14	25	U	28.7/58.6	19.7/26.1	17.9/22.6	32.1/38.2	23.1/64.4	18.9/6.14	17.3	22.3	22.5	22.7	33
76165-constraints_18_0_5_2.sanitized	24	U	360/1030	359/336	333/315	408/378	348/774	350/3400	465	720	382	691	380
76895-cliquecolouring_n13_k8_c7.sanitized	24	U	0.068/0.241	TO	4070/8000	TO	0.066/0.197	4290/1.37*	?	?	?	1410	0.067
76bef-bp4_CB_CSO_LP_FPBEQ_FPBLE.normalised	25	U	308/2160	308/1780	280/1770	382/1970	244/2320	278/5740	418	181	378	320	103
77313-Folkman-175-9054056.sanitized	25	S	98.2	41	40.7	118	59.4	45.3	?	?	?	371	46.7
776ff-as-p43-116-o9-567.sanitized	26	U	786	1300	1650	755	386	1630	110	425	1530	1250	1870
778e0-lec_mult_CvW_12x12.sanitized	24	U	3530/10500	TO	3310/7240	TO	TO	3260/38600	?	?	?	2000	2970
77a0d-usefin_grid_n12_m12	25	U	0.02/0.079	282/725	75.1/234	499/1670	0.116/0.151	0.034/0.036	500	2530	?	118	0.166
77b7f-Folkman-185-19024337.sanitized	24	U	66.6	88.5	140	45.2	45.2	45.2	?	?	218	93.3	86.6
77bfa-case19.normalised	25/26	S	47.7	20.8	22.4	20.8	22.4	40.7	20.6	5.42	3030	4.82	17
77e65-SC25_Timetable_C_495_E_43_CI_35_D_7_T_58.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
780b0-rbsat-v1375c111739gyes10	25	S	44.7	380	TO	786	427	36.4	3450	?	?	45.5	45.5
782b2-vdwb_k6_n300.sanitized	26	S	TO	857	872	TO	TO	887	?	?	?	733	4010
78e8f-RoundRobin_n17_d14	25	U	0.118/0.447	TO	TO	TO	0.417/0.951	TO	?	4820	?	0.117	0.116
79249-grs-64-64	24	U	204/1130	161/633	168/731	175/502	135/778	151/4240	2550	735	276	163	213
79b9e-g2-slp-synthesis-aes-top30	24	S	30.6	16.5	29.4	15.8	26.2	29.6	49.6	43.3	36	30	16
7a044-sum_of_3_cubes_37_bits_87	25	S	164	812	1260	620	363	0.316P	?	399	429	1340	989
7a13f-v19-20.0	26	U	131/377	111/198	138/269	95.6/175	106/340	144/1430	308	204	148	130	204
7a206-oddball_56_5_tto_zp.normalised	25	S	6.73	465	2310	34.3	TO	1.52P	94.8	?	?	2570	5.72
7a44e-Break_triple_12_20.xml	25	S	24.3	3.92	3.42	6.22	14.7	3.62	407	11.1	10.2	8.88	9.68
7a5e9-if-rigid-t2-													

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
83dc8-openstacks-sequencedstrips-nonadd-nonnegated-os-sequencedstrips-p30_3.02	24	U	1360/6200	1260/3540	1090/2930	1280/3030	1330/8980	1740/25700	4660	5000	1270	1580	2120
5-NOTKNOWN													
84282-simon-r20-1.sanitized	25	S	171	0.01	0.01	0.01	0.067	134	?	0.019	0.018	0.066	163
84561-cabp-X-can_715.mtx.rnd-k118.sanitized	26	S	4020	TO	TO	TO	TO	TO	?	?	4860	4180	3860
847c8-ramsey_4_4_19.normalised	25	U	0.168/7.97	TO	TO	TO	207/136	0.166/0.583	?	?	?	?	?
847df-valves-gates-1-k617-unsat.shuffled-as.sat03-412	25	U	929/1330	663/7160	699/9730	1310/12200	806/14900	762/12600	3550	2970	1090	790	783
84995-circuit_48in64out_with_800gates_4in4out_dist128_seed3.sanitized	25	S	8.33	7.93	12.3	9.63	7.28	1.97	44.3	9.43	10.6	7.83	9.98
84c6d-128_100.sanitized	24	S	97.2	69	151	153	103	162	4070	?	111	74.1	91.8
8530c-constraints_16_0_4_1.sanitized	24	S	67.1/023	59.3/88.2	51.1/08.2	66.3/105	61.6/177	59.7/6	158	109	90	68.6	93
85ba2-seed2_0x1e3-216	25	S	43.4	311	206	205	124	0.216	?	78.5	128	340	788
86078-mexam_17_15_2.sanitized	26	U	0.768/1.78	TO	TO	TO	0.717/1.65	TO	?	?	?	0.717	0.717
86173-ktf_TF-7.tf_3_0.06_113	25	S	15.4	25.3	16.2	5.57	15.9	0.266	115	16.5	9.13	26.7	34.4
87040-mchess_16	24	U	11.1/77.9	196/351	271/444	130/81.6	174/483	0.116/0.02	128	204	190	191	0.01
8707e-clqcl_40_6_5.normalised	25	U	0.218/1.3	TO	TO	TO	0.116/0.516	0.116/2.08	?	?	?	0.166	0.166
8733d-reconf10_70_queen14_2	25	S	604	121	142	381	273	147	85.7	3670	328	384	582
87378-hwmccl17miters-xits-iso-6s299b685.sanitized	25	U	1.82/13.4	1.27/12.4*	1.22/12.4*	1.17/12.7*	41.1/5.66*	1.87/66	2.62	2.72	2.42	1.47	1.67
878bc-linked_list_swap_contents_safety_unwind45	24/26	U	41.3/745	27/418	43.7/9570	45.1/8640	50.1/1240	7.93/126	84.8	322	76.5	50.1	40.5
87dbc-ntl-90d-38.sanitized	26	S	9.73	3080	1040	633	1010	1260	730	44.7	763	3040	10.3
87f09-1-ET-512-K-102.sanitized	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
88849-ncc_none_2_18_8_3_1_0_435991723	25	S	642	411	927	1320	817	2.97*	145	2670	?	611	602
88d15-T105_2.0	24	U	330/2680	273/3760	261/5360	264/5450	261/567	237/4380	266	705	400	334	334
88e89-32_200.sanitized	24	S	51.4	37.8	76.2	77	55	87.2	4060	?	64.5	397	49.6
89002-lockchart-group1-L210-K303-p8d4j1.normalised	25	S	546	749	3150	643	TO	146*	?	?	?	570	540
8942d-circuit_32in32out_with_64gates_8in6out_dist128_seed2.sanitized	24	S	16.8	378	377	110	365	5.02	?	?	392	521	16.3
897ac-Nb13T165	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
8990b-BubbleVsPancakeSort_8_5	26	U	1280/3900	1380/2530	1250/2410	2000/3590	1320/4680	1840/12300	?	?	3080	1600	1940
8a05f-arles_thres10_p10_r7466	25	U	0.019/0.254	0.01/0.099	0.009/0.099	0.009/0.099	0.034/0.093	0.066/0.052	0.018	0.018	0.018	0.019	0.018
8a797-GP_300_180_30	25	S	6.53	3.47	3.42	3.42	20.9	3.07*	1610	5.17	5.27	3.67	6.08
8aa9f-adv_gc_n300_k12_UNSAT_s42.sanitized	26	U	1370/6180	165/521	132/363	138/353	351/2240	117/1220	?	145	71.3	328	4380
8af30-simon-r23-0.sanitized	25	S	188	0.01	0.009	0.009	0.066	138	?	0.018	0.018	0.066	183
8b18b-x9-11062.sat.sanitized	25	S	64.1	30.8	107	18.8	34.2	107	556	30.3	97.4	33.5	327
8b223-PancakeVsSelectionSort_6_7	25	U	221/679	160/273	202/406	276/451	191/579	368/2380	290	272	287	162	280
8b267-ER_500_20_4.apx_1_DC-AD	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
8b316-stable-300-0.1-20-98765432130020	24	S	7.03	20.2	53	33.8	21.6	54.5	16.8	99.7	43.5	21.8	27.4
8b91c-lec_mult_DvW_12x11.sanitized	24	U	2020/7310	1540/3640	2090/4870	3710/8150	1580/5660	2140/26700	?	?	3940	1920	2090
8bf63-sokoban-p20.sas.cr.25	24	U	2090/7770	1890/4860	1660/5420	1690/6030	1800/10400	1910/45000*	2200	1020	1990	2000	2640
8caa9-1x1x14_1x2x9_pair_4_1_0_2_1_0.sanitized	26	S	963	1370	328	1160	987	102	?	1280	3700	1400	1370
8d0d8-as-p31-I12-o9-56.sanitized	26	S	37.6	37.3	34.3	19.9	12.2	33.5	26.3	14.9	48.5	18.6	18.8
8d2b0-xorshift_r15_113.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
8d2df-FmlaImplyChain_3_7_8.sanitized	24/26	U	3760/623*	TO	TO	TO	TO	TO	?	?	?	4820	?
8d4b5-hwmccl17miters-xits-iso-6s281b35.sanitized	24	U	13.1/505	5.57/325	5.52/295	5.52/325	17.3/5590	6.68/45000*	?	?	1310	9.89	11.3
8da88-toughsat_factoring_inf	26	S	1810	TO	TO	TO	TO	TO	1560	?	?	?	1500
8db99-Medium-result_b14.af_285.sanitized	26	S	1.37	0.517	0.517	0.517	55.1	1.17	0.768	1.57	1.32	1.37	1.27
8elcd-15.sanitized	26	S	0.218	TO	TO	TO	TO	TO	0.116	?	?	0.217	0.217
8el17-PancakeVsSelectionSort_6_8	25	U	319/1020	309/570	320/562	432/687	335/961	945/3350	540	313	367	292	535
8e5da-bwobdm_28_56.sanitized	26	U	2940/6680	2770/4290	2770/4610	TO	2930/6560	2840/26400	?	?	?	3050	?
8e62c-SC25_Timetable_C_393_E_45_CL_26_D_7_T_50.normalised	25	S	40.1	23.9	12.1	24.9	17.3	15.2	12.1	28.2	20.2	40.4	30
8e720-constraints_17_0_4_1.sanitized	24	S	14.4	6.73	2.37	2.22	7.32	12.6	17.9	35.7	22.3	9.16	6.83
8e811-16_booth_dadda_origin_and_and_dadda_origin_bit28	25	U	1330/3470	1220/2300	1420/2490	TO	1290/3430	1450/13100	?	?	3580	1270	?
8f391-oddball_20_5_tf.normalised	25	U	6.18/13	1.92/4.89	5.78/6.39	8.78/7.2	5.19/9.38	6.38/55.9	16.1	6.98	1.67	7.03	5.42
8f5bc-sgp_7-4-6.shuffled-as.sat05-2671	26	S	TO	TO	TO	TO	TO	TO	?	346	?	?	?
8f8cde-chnl-020x061.sanitized	26	U	0.217/3.6	TO	TO	TO	TO	0.117/3.21	TO	?	?	0.167	0.167
8f8fd-mdp-32-10.sat	25	S	0.117	81.2	1460	1880	89	0.316	?	?	1300	0.117	0.117
901fa-SC25_Timetable_C_492_E_48_CL_33_D_7_T_58.normalised	25	S	TO	553	39	317	TO	0.066	?	811	867	643	1100
9078e-13.sanitized	26	S	0.217	TO	TO	TO	TO	TO	?	?	?	0.217	0.217
907e7-manol-pipe-g10bid_i	24	U	162/355	103/102	161/68.4	171/71	100/147	150/583	566	649	898	110	137
90843-MVRoundRobin_n20_d10_v2	25	U	7.33/72.4	3630/5980	TO	TO	3.68/126	TO	?	?	?	6.98	7.08
90937-mp1-Nb7T08	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
90bec-Wallace_Bits_Fast_8.cnf	25	S	0.769	0.667	0.316	0.316	0.917	0.116	40.1	2.77	1.87	0.768	0.767
9118e-ramsey_3_6_18.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
912ce-hwmccl20miters-iso-mul2.sanitized	24	U	0.818/4.41	0.266/1.43	0.266/1.44	0.266/1.44	0.918/1.75	0.467/4.49	?	?	?	0.768	0.819
915a2-x9-09004.sat.sanitized	24	U	31.2/72.6	31.2/41.8	26.3/36	45.6/58.4	35.2/98.6	27.1/166	148	141	31.7	33.8	56.6
91860-41-119494	26	S	68.5	30.4	22.2	38	47.4	6.12	16.1	8.13	36.5	148	139
918d9-SCPC-500-1	25	U	36.6/78.9	45.9/45.5*	36.2/42*	60.6/60.5*	52.3/129	28.2/11.8*	38	50.5	51.8	52	76.9
91c42-16_16_booth_wallace_mapped_and_and_wallace_origin_bit28	25	U	938/2360	774/1460	798/1530	2970/4370	813/2120	804/8350	3350	2990	1150	801	2540
91e0d-steD5_0x1e3-20	24/26	S	4950	TO	TO	937	TO	0.116	?	?	?	?	?
920bb-SCPC-500-19	26	U	20.7/44.5	19.1/24.7	17.6/23	26.9/31.7	22.4/58.2	18.6/5.74	22.6	25.5	24.7	22.1	32
9276c-combined-crypto1-wff-seed-102-wffvars-500-cryptocplx-31-overlap-2	25	S	22.8	68.2	8.53	161	128	8.93	?	28.2	37.7	23.7	48.9
9277f-bp4_BC012_CSO_FPBEQ_FPBLE_ZR.normalised	25	S	445	520	443	600	2590	454	4360	4880	1140	1190	398
92b22-exam_90_70.sanitized	26	U	0.969/54.3	TO	TO	TO	0.519/36.8	TO	?	?	?	0.918	0.868
93162-DLTM_twitter845_79_19	25	S	86.9	30.4	6.53	15	130	19.7	208	63.9	2.02	35.2	29.1
932e2-bdodo_28_6.sanitized	26	U	3820/8710	2920/5020	3310/5230	TO	3260/8270	3440/36300	?	?	?	3010	?
9376a-cabp-X-can_715.mtx.rnd-k102.sanitized	26	S	138	189	459	333	400	391	419	716	2090	148	138
93983-shuffling-1-s1719683924-of-bench-sat04-310.used-as.sat04-467	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
944b4-b14	25	U	4.22/10.7	3.07/3.19	3.02/4.24	2.97/2.84	3.27/6.31	3.12/16.2	16.8	9.23	5.42	3.87	4.37
94571-af-synthesis_stb_50_200_0_unsat	24	U	4131/1450	444/999	425/1100	657/1330	457/1550	408/5120	468	476	714	422	368
94dd2-Break-X-unsat_18_31.xmz	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
9541d-cabp-V-nos6.mtx.rnd-k275.sanitized	26	S	236	286	459	111	514	276	352	1810	245	230	
958fd-crusti_g2io_200_0.1_127_14.af_151.sanitized	26	U	90.1/428	38.6/225	94/261	93.1/329	40.6/75.2	95.8/369	9.28	20.8	39.8	55	90.3
95931-SC25_Timetable_C_496_E_48_CL_33_D_7_T_50.normalised	25	S	495	TO	3400	2190	265	267	?	903	4120	365	519
95abe-reconf10_68_queen14_1	25	S	92.6	270	187	201	64.8	199	230	3890	164	239	90.8
95f34-quad_res_r30_m35	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
965bc-SCPC-500-5	25	U	11.6/24.1	11.3/13.8	11.6/13.6	18.1/19.2	12.7/33.6	15.5/3.89	10.2	9.98	10.1	13	25.1
965ca-os_fwalk_12	26	U	0.037/0.24	1.07/0.9	1.57/3.34	1.167/3.19	0.0340/1.44	0.162/2.24	?	348	1.52	0.036	0.034
967b5-dubois50.cnf.mis-99.debugged	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
96977-lockchart-group3-L13-K26-p4d3j1.normalised	25	U	2100/6150	2210/3360	2800/4370	3520/5520	1650/5700	2820/35700	3350	4960	2870	1670	1810
96a47-crusti_g2io_175_0.2_511_48.normalised	26	S	280	109	297	310	117	321	2.67	222	37.8	145	295
96a67-WS_500_16_70_10.apx_1_DC-ST	26	U	227/562	212/318	189/282	356/536	210/670	188/1660	362	286	227	228	417
96dea-bv_ILA_Piccolo_JALR_sanitary_transition													

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
a08c1-xorshift_r14_110.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
a08d0-bwmdm_28_2.sanitized	26	U	2350/5810	2160/3930	2080/3730	TO	2340/6060	2210/23900	?	?	?	3950	2220
a08e6-j3045_4_gmto_b	24	S	428	410	401	24.8	46.8	389	3.97	702	194	281	92.8
a09eb-RoundRobin_n18_d16	25	U	0.117/0.648	TO	TO	TO	0.417/0.95	TO	?	?	?	?	0.116/0.117
a0bcd-arles_thres10_p10_r8186	25	U	0.018/0.296	0.009/0.099	0.009/0.099	0.009/0.099	0.0340/0.093	0.066/0.052	0.018	0.019	0.019	0.018	0.018
a110c-xorshift_r14_7.sanitized	26	S	TO	TO	4560	TO	TO	4430	?	?	?	?	?
a14d9-64_200.sanitized	24	S	96.0	69.2	141	143	108	157	?	?	112	70.6	89.2
a1fd1-oddball_22_5_tf.normalised	25	U	11.6/49.5	2.87/8.09	12.3/31.8	16/38.5	3.43/11.4	13.2/32	11.5	6.98	2.62	30.3	20.1
a20f0-vdwb_k5_n260.sanitized	26	U	24.2/83	20/33.1	20.2/33.1	29.2/46.9	21.5/87	20.6/119	19.4	25.2	27	21.6	39.5
a28f5-pj2008_k200	25	S	754	767	571	463	596	552	?	?	2250	681	724
a2ca1-58_134003	25	S	193	719	979	75.3	1070	221	?	325	727	809	798
a2fdd-54_138766	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
a2fe3-mpi1-blockpuzzle_9x9_s1_free9	25	S	27.2	8.33	3.82	41.5	3.88	4.52	2420	1660	45.1	7.88	8.73
a38af-fib65-12-2.used-as.sat04-874	24	S	0.568	4.47	10.5	31.5	5.19	1.02 ^p	1.12	37	0.517	0.567	0.567
a4223-soelberg_unit_223	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
a45a0-x9-10014.sat.sanitized	24	S	53.9	46	27.2	9.08	50	28.1	55.4	8.93	22.3	48.9	67.6
a45ae-Circuit_multiplier36	26	S	3830	TO	3110	TO	TO	3110	?	?	?	3550	1800
a497d-x9-06099.sat.sanitized	25	S	0.517	0.117	0.367	0.417	0.117	0.417	0.367	0.117	0.318	0.117	0.267
a49ec-grs-32-128	25	U	501/2410	408/1540	304/973	518/1630	446/2200	350/8700	?	4740	3970	449	389
a4eec-1x1x13_1x3x6_pair_2_2_1_3_2_0.sanitized	26	S	4520	TO	TO	4340	3550	TO	?	3960	?	?	?
a4f81-bwmdm_28_5.sanitized	26	U	2630/6100	2720/4520	2700/4830	TO	2710/7550	2750/26000	?	?	?	2650	?
a5419-asconhashv12_opt64_H11_M2-IBi5i1RIGz_m0_1_U23.c	24	U	221/593	266/656	305/923	380/613	248/590	489/3820	?	947	260	265	228
a552a-IBM_FV_2004_rule_batch_1_31_1_SAT_dat.k40.debugged	24	U	182/476	172/278	148/233	214/268	179/554	151/1410	?	1220	483	171	192
a58b8-stb_495_168.apx_1_DC-AD	24	U	170/553	132/238	131/219	206/349	181/699	135/1270	212	181	155	149	272
a5dc0-SDP_118_16_5515.sanitized	26	S	TO	950	TO	TO	3950	TO	?	32.1	4250	?	?
a5dc6-mdp-36-10-sat	24	S	0.67	TO	TO	TO	TO	0.717	?	?	?	0.568	0.568
a5fc1-Kakuro-easy-097-ext.xml.hg_4	24	U	1660/610	1480/277	1450/279	1580/278	976/284	1550/5100	778	1290	2000	1820	1580
a60a1-xorshift_r15_191.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
a6ed6-1-ET-256-K-55.sanitized	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
a7088-gm28sparc	25	U	4.07/5.72	10.3/1.68	1.17/1.58	1.17/1.58	6.83/1.05	0.517/2.08	0.317	25.1	2.72	9.57	3.97
a7860-as-p61-122-o8-II-568.sanitized	26	S	TO	TO	TO	TO	TO	TO	?	?	?	?	1900
a789d-13.normalised	26	S	11.6	5.37	5.22	5.32	39.5	2.82 ^p	486	14	7.13	4.69	10.3
a7e5d-mexam_20_18_2.sanitized	26	U	1.87/4.4	TO	TO	TO	1.72/4.24	TO	?	?	?	1.73	1.72
a7e7d-b22_1	26	U	14/26.8	8.68/6.49	9.48/9.7	11.29/45	8.52/10.4	9.63/32.2	35.1	46.1	12	11.3	14
a80a2-bdmdm_28_1.sanitized	26	U	2740/6870	2630/5090	2540/4550	TO	2710/7920	2760/29200	?	?	?	2670	?
a8608-rbsat-v945c61409g3	25	S	132	TO	TO	3870	TO	112	4710	?	2690	134	134
a8dca-asconhashv12_opt64_H10_M2-BPHqhzNzqi_m5_6_U14.c	24	U	278/598	274/631	228/544	359/567	225/601	371/3810	?	841	243	229	246
a917b-1-ZC-512-K-67.sanitized	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
a932c-16_16_booth_dadda_origin_and_and_dadda_origin_bit29	26	U	554/1530	436/835	479/829	1390/2060	469/1330	505/5530	2870	1920	848	459	1750
a957f-T90.2.0	26	U	335/2770	254/3050	289/5190	358/5230	276/490	272/5140	252	898	454	319	334
a95f0-chnl-030x061.sanitized	26	U	0.519/22.4	TO	TO	TO	0.116/21.5	TO	?	?	?	0.267	0.368
a9856-SDP_98_14_7360.sanitized	26	S	11.4	328	74.6	118	129	74.2	358	?	51.1	301	418
a9cb7-mdp-32-16-sat	24	S	0.117	1290	285	366	1390	0.316	?	203	117	0.117	0.167
aa0ea-1x1x13_1x3x6_pair_4_2_0_2_0_0.sanitized	26	S	587	788	1370	1350	2100	786	1150	?	?	705	637
aa2e0-cabp-V-nos6.mtx.rnd-k333.sanitized	?	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
aa39c-urq45	24	U	0.036/0.107	119/659	146/879	101/498	0.034/0.093	0.066/0.036	25	126	133	0.034	0.034
aa4e8-exam_75_70.sanitized	25	?	0.669/41.9	TO	TO	TO	0.266/28	TO	?	?	?	0.668	0.618
aa588-st_815_74_9_2860.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
aa62e-as-p87-130-o6-rot-II-5678-i.sanitized	26	S	311	232	160	204	316	187	210	244	522	249	279
aa6e6-1-ZC-1024-K-117.sanitized	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
aa9b6-x9-10002.sat.sanitized	24	U	177/360	172/221	143/186	286/383	187/564	145/1030	1420	1140	193	186	316
aaa36-linked_list_swap_contents_safety_unwind62	24	U	84.3/1640	135/5700	61.8/15900	97.8/2400	66/2460	16/288	148	518	102	127	83.9
aacfb-barman-pfile06-022.sas.ex.7	24	U	3.97/22.2	0.567/3.73	1.22/30	1.27/32.2	3.53/11.1	2.62/35.2	0.568	91.9	1.97	3.97	4.17
ab02c-sum_of_three_cubes_906_known_representation	25	S	19.5	163	169	118	60.1	1.27 ^p	?	?	613	18.4	18.6
ab17e-vdwb_k6_n250.sanitized	26	S	141	305	320	215	801	314	?	?	946	286	293
ab2ab-1.sanitized	26	S	0.218	TO	TO	TO	TO	0.067	?	?	?	0.217	0.217
ab343-mdp-36-14-sat	24	S	0.618	TO	TO	TO	TO	0.717	?	?	?	0.568	0.567
abada-arles_thres10_p20_r4305	25	U	0.019/0.377	0.01/0.181	0.009/0.182	0.009/0.181	0.034/0.11	0.066/0.1	0.018	0.018	0.018	0.019	0.019
ac347-cliquecolouring_n41_k5_c4.sanitized	24	U	1.12/1.71	TO	TO	TO	0.116/0.366	TO	?	?	?	1.12	1.12
ac388-pj2008_k80	25	S	163	106	81.9	164	65	83.8	?	3610	2720	281	161
ac3e3-E00X23	26	U	1720/7180	1210/6130	1410/5730	973/4780	1050/2880	1540/19400	?	2530	2480	1510	1490
ac51d-WS_500_16_90_70.apx_1_DC-ST	25	S	28	89	24.2	26.9	88.7	24.8	45.7	53.9	25	96	73.5
ac695-SC21_Timetable_C_527_E_71_CL_35_S_35	26	S	56.3	30.6	36	34.8	62.9	39.9	94.5	112	125	44.4	55.4
ac9f1-1.sanitized	26	S	0.218	TO	TO	TO	TO	0.066	?	?	?	0.217	0.218
acc0b-BubbleVsPancakeSort_8_6	25	U	1980/550	3340/5480	1740/2530	2600/3320	2220/6340	2390/19600	?	?	?	3360	3000
aceab-lra_8.sanitized	24	S	29.7	19.2	18.8	19.1	50.1	17.8	25.7	49.4	21.4	19.2	29.1
ad224-schoolt-5-3-9-2-4-5-10-0-1-0-2-seed3.sanitized	26	U	173/730	91.8/167	100/184	103/167	78.8/472	99.5/87	?	?	232	182	186
ad4e1-x9-08075.sat.sanitized	24	U	14.3/32.3	10.5/13.1	11.9/15.7	15.7/17.1	11.6/27.8	12.1/68.9	22.4	21.7	11.1	11.6	20.7
ad5b8-xorshift_r15_104.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
ad69e-g2_T49.2.0	25	U	1230/2850	731/649	738/4650	874/3980	796/151	831/9310	1900	2900	2660	797	1250
ad6bc-as-p99-133-o6-rot-rfl-568-i.sanitized	26	S	353	304	139	206	135	143	323	223	344	349	304
ad9eb-AProVE07-21	25	U	1.57/8.35	1.17/0.482	0.867/0.632	0.717/0.282	0.967/0.352	0.967/4.69	49.9	27.3	2.92	1.47	1.62
adf6c-circuit_48in64out_with_800gates_4in4out_dist128_seed1.sanitized	24	U	16.4	13.4	13.7	15	17.9	4.37	85.8	22.6	19.8	15.6	19.8
ae48d-korf-17	26	U	45.5/111	35.6/52.3	37.7/48.6	127/174	38.7/92.9	39.3/298	284	135	86.7	39.9	63.1
ae952-srhd-sgi-m37-q446.25-n35-p30-s33692332	26	U	0.368	0.018	0.018	0.018	0.116	0.166 ^p	0.034	0.067	0.034	0.367	0.368
ae9c5-16_16_and_wallace_origin_and_default_mapped_ultra_bit27	25	U	1140/2920	1010/2080	1160/2170	3580/6230	1100/3420	1180/13300	4450	4450	2690	1020	4100
af05a-x9-12098.sat.sanitized	25	S	212	552	426	88	580	444	4260	144	411	567	904
af750-si2-b03m-m800-03	26	S	8.13	2.47	3.07	2.52	4.44	3.52 ^p	1.12	102	15	6.28	7.38
af824-semibuster_4200.af_72.sanitized	24	S	1.68	0.316	0.266	0.316	1.92	0.667	0.418	0.719	0.719	1.57	1.67
af9c2-cabp-T-dwt_592.mtx.rnd-k145.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
af8e-SC25_Timetable_C_498_E_46_CL_34_D_7_T_50.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
b0142-count_p2_M21.sanitized	26	U	0.117/0.237	TO	TO	TO	TO	0.117/0.219	?	?	?	0.116	0.116
b023e-mpi-tr1_ali_s11_c35_bail_UNs	26	U	42.5/957	30.6/43.9	38.3/57.3	49.3/64.9	32/92.2	38.9/320	48.2	89.9	36.9	31.6	44.5
b087c-Break_unsat_14_23.xml	26	U	TO	TO	TO	TO	TO	TO	?	?	?	?	?
b087c-Break_unsat_14_23.xml	26	U	220/938	71.8/127	166/271	218/179	60.3/328	168/1850	?	?	43	141	143
b087c-Break_unsat_14_23.xml	26	U	734/2402	766/1270	753/1270	1380/2330	670/2760	767/8480	1490	1290	768	712	1530
b1137-stb_418_125.apx_2_DC-AD	26	U	586/2140	447/1190	559/1450	561/2400	558/5590	?	?	?	841	542	521
b13ba-adv_gc_n100_k14_UNSAT_s42.sanitized	26	U	4140/10000	4200/5620	4050/5360	TO	3950/11000	3990/1.42 ^p	?	?	?	4060	?
b172b-Schur_161_5_d38	24	U	TO	TO	TO	TO	TO	TO	?	?	?	?	?
b18b2-div_miter_lec_2	25	?	TO	TO	TO	TO	TO						

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
bdcdb-lec_mult_CvK_11x10.sanitized	24	U	396/1170	332/757	329/763	732/1550	362/1120	336/3170	1850	2830	573	354	874
be188-af-synthesis_stb_50_40_9_sat	24	S	364	510	89.7	255	293	484	1060	553	643	774	988
be641-j3037_10_mdd_b	24	S	27.8	18.6	10.8	51.1	35.8	10.9	249	22.4	6.32	37.1	2.77
be78a-4.normalised	25	S	12.5	5.98	5.93	5.82	43.1	3.17P	301	14.6	7.1	6.73	11.1
bea83-xor_op_n36_d3	25	U	78.7/208	2120/2180	2010/2140	1870/1600	60.3/260	2570/9400	?	?	2610	104	86.8
b78c8-sec_6087_19_0.3_0.1_20.af_130.sanitized	26	U	12/91.2	9.29/95.9	10.9/103	24.9/115	8.43/55.5	9.93/76.7	4.67	7.88	73.5	13.4	35
b7add-pcmx-scheduling-m19-10199-62102-UNSAT.sanitized	24	U	4880/17100	4090/8420	TO	3930/7210	2390/11900	TO	?	?	?	4810	?
b7c82-snw_17_9_CCSPreOptEncpre	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
c0074-st_659_37_25_686.normalised	25	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
c0293-mp1-Nb7f42	24	S	62.7	TO	TO	61	93.6	2180	?	6.88	33.2	?	201
c0335-reconf10_73_queen13_2	25	S	142	227	118	74.2	90.1	120	294	797	309	40.9	141
c03f7-linked_list_swap_contents_safety_unwind68	24	U	262/10800	87.9/1260	56.2/16500	60.4/17200	141/4380	19.2/373	217	599	138	114	249
c05d4-MVRoundRobin_n16_d10_v2	25	U	3.02/25.4	TO	TO	TO	1.52/39.6	TO	?	?	?	2.97	3.02
c0bd8-goldcrest-and-14	25	U	678/11100	686/12900	698/9450	787/9450	576/13400	729/19800	?	?	2130	724	735
c0e6e-x9-10093.sat.sanitized	25	U	100/218	90.1/125	85.8/120	117/159	101/304	87.4/547	551	390	119	97.2	181
c0e82-lec_mult_CvW_11x10.sanitized	25	U	236/691	147/304	194/396	249/526	161/484	200/1860	734	911	270	161	236
c14f0-vdw_3_13_n161.sanitized	26	U	563/2720	468/1070	467/1050	754/1810	497/2530	508/7850	1020	715	693	461	524
c15b0-snw_16_9_preOpt_pre	26	S	314	85.3	1800	268	89.1	1.52P	197	247	540	97.1	716
c2109-case17.normalised	25	S	227	25.1	20.3	82.4	27.7	20.9	399	1550	185	27.3	109
c214f-64_150.sanitized	24	S	72.5	52	106	108	75.5	117	4810	2700	85.6	55.8	67.9
c21ce-linked_list_swap_contents_safety_unwind70	24	U	58.3/2170	50.6/1460	157/40000*	76.1/24100	58.6/4030	21.3/419	261	647	112	66.8	52.6
c2619-manthey_single-ordered-initialized-w48-b9	24	U	548/976	485/635	480/635	805/970	551/1370	457/5080	893	683	548	496	749
c2bfe-frb45-21_2.used-as.sat04-884	24	S	29.7	89.5	229	207	97.3	37.4P	287	270	47.3	30.4	30.4
c3004-cabp-M-bcsstk06.mtx.rnd-k36.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
c3bc9-2.xz.sanitized	26	S	6.13	3.97	3.92	3.92	23.3	4.22	3.67	10.5	4.82	3.77	5.62
c3e53-x9-09007.sat.sanitized	24	U	21.6/48.5	20.6/62.3	17.6/22.7	27.4/32	22.9/59.3	18/107	117	62.2	25.6	22.8	33.3
c4476-mexam_14_12_2.sanitized	26	U	0.317/0.645	TO	TO	TO	0.266/0.598	TO	?	481	?	0.267	0.267
c557a-sec_7216_12_0.5_0.2_11.af_463.sanitized	26	U	11.9/304	7.13/293	26.8/293	7.48/262	7.13/292	9.18/159	3.42	44.4	149	11.1	12.2
c569d-oddball_13_5_ttf.normalised	25	U	22.8/138	18.9/65.8	15.2/49.8	6.18/18.4	18.9/165	16/192	37.3	6.72	1.22	25.4	24.7
c5a98-x9-11053.sat.sanitized	24	S	29.4	120	32.6	21	131	33.1	2690	2400	15.1	128	67.5
c5ae0-Circuit_multiplier22	24	S	33.4	570	284	227	595	291	496	4480	104	381	389
c615d-preimage_80r_486m_160h_seed_306	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
c62a1-1x1x14_1x2x9_pair_2_2_0_3_2_1.sanitized	25	S	3090	2920	3250	3430	2840	4310	?	1380	3720	3010	3920
c6470-bp4_IXA_FPBEQ_ZR.normalised	25	S	513	252	401	228	299	418	4490	911	89.3	596	175
c6487-genurq7Sat.shuffled-as.sat03-1513.cnf.mis-135.debugged	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
c6568-x9-10076.sat.sanitized	24	U	76.6/172	88.4/117	62.7/85	87/116	97.3/270	63.7/395	350	?	304	86.5	95.8
c69e5-iseitin_d3_n162.sanitized	24	U	0.02/0.062	TO	TO	TO	0.066/0.093	0.066/0.026	?	?	?	0.117	0.019
c6e03-hwb-n24-02-5786928571.shuffled-as.sat03-1618	24	U	105/325	88.1/191	98.5/214	91.1/203	94.1/358	99.8/817	166	248	139	95.5	141
c73ed-noL_11-12.sanitized	24	S	180	516	224	123	165	0.034	?	?	403	116	119
c7449-bdoavm_28_4.sanitized	26	U	3440/7990	2560/4260	2980/5290	TO	2680/6740	2940/33600	?	?	?	2580	?
c76f13-mp1-blockpuzzle_5x10_s7_free4	24	U	8.03/30.3	9.13/20	10.7/24.9	11.4/24.1	4.97/28.4	11.29/3.8	466	164	10.3	6.42	6.63
c7ee9-SC25_Timetable_C_481_E_48_Cl_32_D_7_T_58.normalised	24	S	4.48	1023	25.3	110	25.4	79.3	69.5	82.3	16.6	62.2	231
c801a-worker_20_40_20_0.95	24	S	0.418	0.066	0.066	0.066	0.817	0.116	0.116	0.117	0.367	0.368	?
c82ee-preimage_80r_495m_160h_seed_379	24	S	5.83	689	1160	833	697	1010	334	541	202	620	319
c86b2-TTC2021_Middle_9.xml	25	S	9.23	6.22	3.47	3.32	12.2	6.74	0.567	46.9	0.868	8.13	3.57
c8b12-linked_list_swap_contents_safety_unwind78	24	U	150/3020	125/9710	75/25600	157/40000*	155/17000	26.9/528	253	748	93.6	109	156
c8cde-bp5_CSO.normalised	25/26	S	272	604	190	332	216	179	?	763	732	698	482
c8d21-cliquecoloring_n14_k7_c6	25	U	14.9/53.9	1420/2960	1010/2080	2370/3070	0.166/0.266	939/11600	?	3110	417	14	14.7
c8e64-64_25.sanitized	24	S	10.8	8.08	17.8	17.6	27.3	20.6	74.2	437	13.7	8.03	10.3
c9f51-af-synthesis_stb_50_140_3_unsat	24	U	156/551	160/360	145/416	196/511	171/556	140/1610	114	141	323	124	138
ca14a-ITC2021_Early_3.xml	24	S	2.48	0.217	0.468	0.468	16.5	0.567	0.067	2.22	1.17	2.12	2.37
ca18f-as-p75-127-06-rot-4568.sanitized	24	S	TO	TO	TO	TO	TO	4490	?	?	?	?	1850
ca3c6-case16.normalised	24	S	184	229	121	96.3	250	126	1160	?	191	244	182
ca717-xorshift_r16_180.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
cabfd-oddball_17_5_ttf.normalised	25	U	3.93/5.89	0.517/1.53	1.07/1.88	1.27/2.24	0.767/2.75	1.32/8.14	3.13	1.97	0.721	3.52	3.87
cb2e8-clique_n2_k10.sanitized	24/26	U	0.118/0.239	6.12/16.8	8.23/24.2	14.3/36.9	0.116/0.199	0.066/0.019	3590	301	10.3	0.117	0.116
cb262-RoundRobin_n17_d13	25	U	0.119/0.387	4380/2510	TO	TO	0.366/0.516	TO	?	625	?	0.067	0.066
cb950-rhpb3_050_shuffled	25	U	0.168/3.37	TO	TO	TO	0.066/3.17	0.116/0.1	?	?	?	0.117	0.116
cbd09-em_11_3_4_cmp	25	S	34.9	100	167	243	53.5	0.166	205	45.8	3	305	300
cc30c-j3037_9_mdd_bm1	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
cc42c-g2_T97.2_1	26	U	2020/6850	2320/8540	2350/14900	3250/13700	2550/1410	2230/27200	?	?	?	2860	2100
cc436-minandmaxor128	26	U	588/1300	386/771	627/722	480/772	451/918	618/8810	?	4270	2700	427	648
ccc66-VanDerWaerden_pd_2-3-23_505	25	S	956	715	78.1	731	737	21.7P	404	1910	1330	699	1490
cccee-1-ET-512-K-96.sanitized	24	S	0.518	TO	1900	409	4610	0.867	826	4580	3970	?	0.518
cd33a-MASG0_72_keystream76_0	24	S	1880	3060	815	350	548	816	14.9	4110	3740	3150	505
cd361-ecarev-110-4099-22-30-7	24	S	380	213	243	210	207	265	217	279	296	219	283
cdce6-simon-r17-0.sanitized	24	S	153	0.01	0.01	0.009	0.066	128	?	0.018	0.018	0.066	148
cdd13-6g_6color_366_050_04	24/25	S	TO	TO	2420	164	17	3810	380	1550	?	3640	801
cdd89-sted2_0x0_n219-342	25	S	1240	119	1110	456	1130	0.367	364	2430	1680	131	2720
cdf55-at-least-two-traffic_kkb_unknown	25	U	288/346	218/456	223/457	196/498	156/361	238/3960	83.3	1270	?	198	289
d0298-Break_triple_16_70.xml	25	S	34.9	29.7	169	45	58.1	12.7	277	80.3	59.8	89	18.6
d02fa-exam_flat_0.04_2018_3	24	S	937	663	404	711	293	432	?	3650	2720	497	856
d07b7-544707209399nc.shuffled-as.sat03-1670	24	S	13.4	12.2	11.3	37.7	24.7	0.617	14.8	32.5	27.7	12.2	24.4
d0b5d-connm-ue-csp-sat-n600-0.04-s1793042357.used-as.sat04-975	26	U	25.1/174.4	19/35	18.7/35.4	27.6/44.4	20.3/54	19.1/319	31.7	34.9	34.8	20.2	22.7
d10b8-xorshift_r15_175.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
d1194-mdp-32-14-sat	24	S	0.469	3480	TO	TO	2180	3690	0.667	?	?	0.367	0.367
d1335-mulso1_1.48.sanitized	26	U	2.17/255	TO	TO	TO	0.267/24.9	TO	?	?	?	0.568	0.568
d1940-mdp-28-11-sat	24	U	0.036	130	271	196	141	0.060	1030	368	168	39.3	0.034
d1954-x9-09098.sat.sanitized	24	U	48.9/106	41.7/53.6	39/50.2	48.7/61	46.2/123	39.9/235	183	142	16.3	45.6	54.1
d1c28-aries_thres10_p10_r8175	25	U	0.019/0.252	0.01/0.1	0.01/0.099	0.009/0.099	0.034/0.094	0.066/0.052	0.018	0.019	0.018	0.018	0.018
d27f7-exam_95_70.sanitized	26	U	0.919/57.8	TO	TO	TO	0.418/39.2	TO	?	?	?	0.969	0.868
d2af3-lockchart-group1-L190-K276-p84djl.normalised	25	S	538	754	1440	661	665	59.6P	?	1290	?	602	524
d35df-16_2	25	S	32.6	TO	TO	TO	TO	22.8	?	?	?	?	30.9
d3893-g2-ak128boothbg2msaig	24	S	1.52	0.367	0.366	0.367	1.97	1.12	?	1.12	0.468	1.22	1.37
d3e4e-bp4_BC012_AM_IXA_LPI.normalised	25	U	286/431	228/304	210/373	299/229	247/614	211/10500	?	?	273	207	318
d40a6-Folkman-180-5383714.sanitized	24	S	133	123	42.2	113	62.2	41.8	?	?	139	126	87.2
d4186-xorshift_r14_85.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
d4219-snw_16_8_preOpt_pre	25	U	115/1520	96.4/2390	107/2290	99.1/2170	104/1150	284/922	124	258	233	109	115
d4256-PancakeVsSelectionSort_6_6	25	U	80/289	83.1/160	65.3/137	111/205	71.8/240	159/1080	2				

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
ddf96-oddball_57_5_to_zp.normalised	25	S	7.22	4700	TO	110	1400	1.52P	467	?	?	?	5.93
de344-SDP_136_18_712.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
de7a6-nol-11-18.sanitized	24	S	31.9	56	97.5	89.8	34.1	0.266	?	1160	37.9	126	131
de898-sudoku-N30-16	25	U	2720/11400	2610/9540	1840/12000	1870/10900	2540/6190	2080/4800	?	?	?	?	2660
de8c4-16_16_booth_dadda_mapped_and_and_wallace_origin_bit28	25/26	U	955/2750	912/1690	939/1770	3970/5490	961/2600	933/9640	4120	4000	1360	970	3640
ded23-x9-09014.sat.sanitized	24	U	27.2/60.2	22.6/29.3	23.6/31.4	51.4/65.7	25.1/65.4	24.2/145	116	95.3	27.7	25	42.5
df1bd-x9-12021.sat.sanitized	24	S	117	99.6	222	679	109	227	?	3590	266	108	165
df813-veer_axi_yosysqh_appnote_123_veer_axi-p06_transition	25	U	9.99/491	3.57/70.9	4.27/83.2	3.97/82.2	12/1030	5.27/392	?	2500	24	6.68	9.38
df84e-nla-digibench-scaling_dijkstra_u_valuebound1_step	25	U	350/1420	306/905	261/1360	323/1580	313/6430	261/18800	?	1900	430	251	359
e0674-af-synthesis_sfb_50_200_0_sat	24	S	96.7	93.9	277	68.6	231	26.5	64.2	16.7	147	163	?
e06be-hwmc20miters-iso-mul7.sanitized	24	U	3.72/18.9	0.967/5.89	0.967/5.84	0.967/5.84	3.78/13	1.57/24.2	?	?	?	3.07	3.52
e08f1-x9-10096.sat.sanitized	24	U	95.6/211	81/109	76.3/107	104/141	88.4/248	78.7/550	400	417	84.1	89.4	123
e0bf2-md5_48_1	24	S	88.3	182	21.9	56.3	323	22.9	206	46.9	112	83.4	222
e1287-rhps5_085_shuffled	25	U	9.13/36.9	TO	TO	TO	0.116/24.8	0.216/0.232	?	?	?	8.03	8.28
e1292-Medium-result_b30_af_80.sanitized	26	S	1080	0.116	0.116	0.116	45.8	0.717	0.167	0.318	0.268	0.969	1190
e207b-fermat-834855329100173267	25	S	0.067	2770	1170	TO	3170	0.116	?	?	?	?	0.067
e23ed-14.normalised	25	S	9.83	4.68	4.68	4.63	34	2.77P	87.7	13.1	5.83	4.43	9.28
e2b81-59-134241	26	S	TO	TO	TO	3800	TO	TO	?	?	?	?	?
e2d2b-59-129706	25	S	1410	551	389	68.6	1010	1070	3710	561	812	526	530
e2deb-circuit_48in24out_with_100gates_7in7out_dist128_seed1.sanitized	24	S	1.07	58.8	56.9	59.8	62.8	3.42	326	301	185	170	1.02
e305d-ntil_90d-35.sanitized	26	S	147	32.5	199	135	457	34.2	495	133	1160	38.7	139
e318e-544707209399nw.shuffled-as.sat03-1671	25	S	37.3	10.1	25.8	16.9	12.8	0.366	30.9	101	392	29.1	54.3
e371c-32_350.sanitized	24	S	92.6	68.8	131	138	101	158	?	?	?	107	74.8
e3860-lec_mult_CvK_11x11.sanitized	24	U	1160/3190	911/2030	1110/2520	2650/5200	978/3140	1110/13800	4980	?	?	1740	946
e3a13-9.sanitized	26	S	0.168	TO	TO	TO	TO	0.116	?	?	?	0.219	0.167
e3bd4-crusti_g2io_250_0_2_255_43.normalised	25	S	212	179	87.4	98.3	178	154	12.7	234	717	236	218
e430a-rhph_p25_r25	25/26	U	0.268/8.95	TO	TO	TO	0.117/9.43	0.166/0.182	?	?	?	0.116	0.118
e4422-sudoku-N30-23	25	U	TO	TO	3330/21700	TO	TO	4040/1.37*	?	?	?	?	?
e481a-j3037_1_gmto_b	24	S	136	50.2	7.58	16.1	46.4	7.68	327	19.3	40.5	65.3	79.4
e4ac1-2013113162201nw.shuffled-as.sat03-1668	25	S	25.3	5.92	40.1	35.8	19.3	0.416	734	176	18.7	5.42	48.6
e4d18-quad_res_r33_m36	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
e5066-1x1x14_1x2x9_pair_2_4_0_3_1_0.sanitized	26	S	2120	1020	1000	TO	3020	480	2600	?	?	1000	4070
e55a4-af-synthesis_sfb_50_120_4_unsat	26	U	406/1440	327/678	333/794	478/846	371/1350	323/3400	382	377	495	325	312
e5787-fpsol2.1.2.29.sanitized	26	U	2.82/68.5	TO	TO	TO	0.267/69.6	?	?	?	?	0.769	0.873
e6747-W5_500_16_70_10.apx_2_DS-ST	26	U	340	202	54.1	340	87.2	55.6	63.7	1150	58.2	222	60
e685a-sbtp4_140_2_sanitized	26	U	2930/7070	2460/4420	2630/4550	TO	2950/8510	2840/33300	?	?	?	2570	?
e6cdc-GracefulGraph-K05-P02_c18	24	S	38.8	35.1	16.7	16.1	5.23	17.2	8.23	103	61.6	27	15.8
e707c-Break_triple_14_48.xml	25	S	87.1	96.2	33.5	34.3	113	307	26	91.2	35.8	30	21
e7248-Folkman-190-35800471.sanitized	24	S	286	588	517	378	382	67.5	?	?	664	519	389
e7add-mpi-klieber2017s-0300-032-t12	25	S	34.3	34.7	39.6	12.7	80.3	39.8	805	38.6	141	39.4	49.1
e7ee7-circuit_32in32out_with_64gates_7in7out_dist128_seed2.sanitized	25	S	0.619	54	50.2	57.9	55	2.42	909	528	284	68.2	0.568
e84b0-ER_400_20_4.apx_2_DC-ST	24	U	196/725	150/292	180/347	197/385	179/855	182/2080	242	236	258	159	212
e85e7-1-ET-512-K-110.sanitized	24	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
e85fb-lru_7.sanitized	24/26	S	26.2	16.6	16.7	16.4	41.9	15.6	20.5	45.9	18.8	15.3	25.3
e8abb-linked_list_swap_contents_safety_unwind57	24	U	205/6740	35.7/823	53.8/14600	75.9/20100	39.4/2060	12.7/216	128	428	67.8	153	192
e8ccb-ntil_90d-36.sanitized	26	S	116	26.6	26	132	104	28.6	54.2	72.7	394	31.9	109
e9125-lattice-20.19.sanitized	26	U	0.918/4.48	TO	TO	TO	2.87/39.6	TO	?	?	?	0.468	0.367
e99e2-rbsat-v760c43649gys9	25	S	5.12	124	83.4	70.5	136	4.47	174	152	87	5.22	5.17
e9f6c-crusti_g2io_250_0_2_255_12.normalised	25	S	342	290	106	243	270	11.8	18.8	299	71.5	344	324
ea37a-clccl_30_9_8.normalised	25	U	0.268/1.68	TO	TO	TO	0.117/0.751	0.166/3.84	?	?	?	0.217	0.267
ea401-sudoku-N30-15	25	U	299/1520	259/1260	201/1150	241/1770	224/3020	?	?	?	944	297	303
ea827-ER_500_30_3.apx_2_DC-AD	24	U	594/2370	476/1190	535/1270	556/1390	597/3450	541/9790	874	1360	745	250	655
ebbda-2018D_VexRiscv-regch0-20-p1_step	25	U	146/4200	135/1980	70.4/507	66.8/482	126/7940	87.8/1130	?	1150	284	185	162
ebc40-MM-23-2-2-3-B	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
ec84e-mchess_15	24	U	9.62/63.9	32.1/57.6	214/373	43.2/65.5	102/361	0.116/0.018	103	366	35.4	120	0.018
ecca7-004	24	S	84.1	109	57	125	24.9	68.4	69.4	77.2	89	116	61.9
ed6d8-battleship-16-31-sat	25	S	0.167	0.216	8.63	5.77	1.17	0.066P	?	5.02	12.4	18.6	19.1
ed9f3-TC2021_Early_9.xml	25	S	2.82	0.867	0.967	0.467	0.467	0.107	0.067	1.92	2.37	3.17	1.37
eda4a-pb_300_05_lb_17	25	S	21.4	15.5	41.2	11.8	48.7	37.4	15.1	45.1	32.3	19.2	12.4
edceb-x9-09057.sat.sanitized	24	U	25.5/55.7	21.2/27.3	21.7/28.3	28.6/35.4	23.1/60.7	22.2/148	85.2	82.1	22.1	23.3	35.3
eddd6-x9-10098.sat.sanitized	24	S	37	5.67	16.6	26.6	6.39	17	23.6	16.8	41.7	6.48	89.5
ee44c-n384p5q2_vh.af_138.sanitized	26	S	6.28	2.42	13.3	4.22	2.83	15.3	1.77	0.417	19.1	2.92	2.97
ee576-oisc-subrv-sll-nested-13	24	U	1410/28800*	1630/40000*	1770/17300	2080/12000	2090/28800*	1760/26.3*	?	?	?	1450	1430
ee5fb-11.normalised	25	S	82.5	29.5	28.5	28.2	73.4	12.5P	?	?	80.8	33.7	29
ee8c8-xz.sanitized	26	S	11	6.83	6.78	6.73	40	6.68	6.44	18.1	8.03	6.58	10.5
eed51-Folkman-180-11710376.sanitized	24	S	134	72.2	41.1	70.6	80.1	45.3	?	?	254	78.5	48.1
ef063-cabp-V-nos6.mtx.rnd.k297.sanitized	26	S	1070	1420	867	737	2400	924	?	?	3980	1190	1090
ef169-Folkman-185-152478531.sanitized	24	S	208	437	180	346	225	208	?	?	351	443	199
ef330-crn_11_99_u	24	U	1.42/3.54	1.02/0.383	1.17/0.382	0.767/0.382	1.070/3.67	0.0340/0.132	0.768	0.719	1.07	1.02	1.37
efc1b-oddball_19_4_ttf.normalised	25	U	40.2/189	45.6/127	55/103	13.5/31.1	55.1/257	58.4/825	8.27	5.42	10	44.3	90.2
f0426-hwmc12miters-xits-iso-6111.sanitized	24	U	6.33/29	1.37/11.9	1.37/11.4	1.42/12.3	9.01/10.2	2.12/488	?	97.9	2.57	5.92	6.48
f0542-summle_X11112_steps6_11-2-2-4-4-8-25-100	24	S	10.8	20.4	9.18	31.1	16.1	0.266	29.4	18.9	16.8	14.3	30.2
f0753-Nb54T6	25	U	TO	TO	4870/40000*	4950/30400	TO	TO	1130	2860	?	?	?
f0fab-fiv-mitem172	25	U	40.2/103	32.9/59.6	31.6/61.8	36.7/60.6	38.2/106	31.6/272	63.3	65	45.5	32.9	35.4
f15a3-crusti_g2io_350_0_5_255_42.af_264.sanitized	26	S	19.5	3.47	3.32	3.22	11.6	3.87	204	18.8	9.58	17.2	18.5
f15ad-ER_400_20_7.apx_1_DC-AD	24	U	444/1630	419/945	386/890	553/1200	453/2550	402/7490	440	815	549	453	549
f16b7-af-synthesis_sfb_50_100_9_unsat	24	U	384/1340	394/762	380/833	593/1230	406/1390	369/3430	421	441	490	407	382
f17df-SC25_Timetable_C_406_E_45_CL_26_D_7_T_50.normalised	24	S	172	25.2	61.9	57.4	157	2140	?	45.9	19.2	34.2	427
f18c7-FmlImplyChain_3_7_7.sanitized	24	U	53.2/45.3	662/463	340/215	711/302	235/176	484/2730	360	211	?	34	35.1
f1973-lockchart-group2-rnd0.3-19-K38-P8D4J1_4	25	S	241	457	TO	184	355	33P	?	367	1350	1680	1740
f1cd6-oddball_24_4_ttf.normalised	25	U	245/937	253/410	235/511	127/282	346/1780	246/3820	17.9	6.83	83.1	306	92.2
f1d1a-openstacks-sequencedstrips-nonadl-nonnegated-os-sequencedstrips-p30_1.035	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
-NOTKNOWN													
f25a1-1-TC-256-K-63.sanitized	25	S	101	112	89.9	41.7	144	0.317	21.3	112	172	209	122
f296f-x9-08014.sat.sanitized	24	U	12.8/28.5	10.8/15.1	10.7/15.2	16/20.9	11/28.6	11/62.5	46.4	30	11.2	12	23.9
f26d-sc9_9630_26_0_3_0_17.af_516.sanitized	26	U	75.5/309	57.8/414	44.9/398	42.7/521	35.8/149	24.1/313	15.7	34.2	32.9	76.5	73.9
f32bb-DLTM_twitter774_83_17	24/25	S	949	261	691	247	261	679	2878	835	94	252	122
f33a6-simon-r17-1.sanitized	24	S	154	0.009	0.009	0.009	0.066	130	?	0.018	0.01		

Benchmark	SC	S/U	T	K4	K24	AE	SK	L	V1	V2	V3	V4	V5
fcc4c-x-epic_a19-p16_step	25	U	300/10500	180/9930	192/1740	179/1850	165/27100	199/4180	?	3710	860	235	294
fd1e0-ntil-90d-42.sanitized	26	S	2600	TO	TO	1670	2280	TO	?	?	?	?	2510
fdc4d-bp4_CB_LP_FPBLE.normalised	25	U	112/847	76.8/596	149/592	226/813	144/892	152/6420	?	352	126	108	131
fdafc-oddball_29_4_ttf.normalised	25	U	3300/13200	968/1500	3370/7610	2320/5650	958/3000	3480/1.32 ^s	682	130	588	1450	1450
fe776-vdwb_k7_n3703.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
fe851-grs-256-64	25	U	1170/9300	1010/7810	1240/10100	925/5250	731/4670 ^s	885/41400	?	4490	963	1160	1020
fe9b9-lockchart-group1-L220-K317-p8d4j1.normalised	25	S	TO	TO	4010	1090	TO	307 ^p	?	?	?	?	?
fea0f-16_16_default_mapped_ultra_and_and_dadda_mapped_bit28	25	U	564/1570	500/991	555/1110	1340/2190	542/1590	559/5430	2180	1880	1030	516	1320
fed1f-ntil-90d-41.sanitized	26	S	TO	TO	2790	4170	TO	TO	?	?	?	?	?
fed3f-8.sanitized	26	S	0.217	TO	TO	TO	TO	0.066	?	?	?	0.217	0.218
ff706-adv_gc_n200_k15_UNSAT_s42.sanitized	26	?	TO	TO	TO	TO	TO	TO	?	?	?	?	?
ff9e9-1-ZC-512-K-61.sanitized	24	S	0.268	333	534	113	279	0.717	147	167	186	300	0.267

C.2 Public-interface overhead

C++ and C compare the production implementation with the public C interface. All 383 pairs are included, with shared outcomes for the two implementations. A, B+C, R, and W identify Tier A, B28+C5, Tier R, and Tier W.

Table 9: Per-instance interface runtimes (read down each half).

Tier	Benchmark	S/U	C++	C	Tier	Benchmark	S/U	C++	C
A	color-n14-k3-s006.cnf	U	0.009	0.008	A	xor-n56-s001.cnf	U	0.007	0.008
A	color-n15-k4-s007.cnf	U	0.008	0.008	A	xor-n65-s005.cnf	U	0.007	0.008
A	color-n16-k4-s009.cnf	U	0.008	0.008	A	xor-n70-s007.cnf	U	0.007	0.021
A	color-n17-k4-s011.cnf	U	0.008	0.023	A	xor-n71-s011.cnf	U	0.007	0.008
A	color-n18-k3-s010.cnf	U	0.008	0.008	A	xor-n84-s000.cnf	S	0.008	0.008
A	color-n26-k4-s001.cnf	U	0.009	0.01	A	xor-n86-s008.cnf	S	0.007	0.01
A	color-n27-k3-s004.cnf	U	0.008	0.009	A	xor-n94-s016.cnf	S	0.009	0.007
A	color-n28-k3-s012.cnf	U	0.008	0.008	A	xor-n96-s019.cnf	U	0.007	0.008
A	color-n29-k3-s002.cnf	U	0.009	0.009	B+C	2013113162201nw.shuffled-as.sat03-1668	S	22.7	22.7
A	color-n32-k4-s003.cnf	U	0.009	0.012	B+C	9vliw_bp_mc.shuffled	U	10.7	10.9
A	color-n33-k4-s005.cnf	U	0.028	0.01	B+C	DecompressReader_read_12	U	0.034	0.067
A	color-n34-k3-s000.cnf	U	0.009	0.009	B+C	DivS_862_11.cnf.sanitized	S	28.2	28.2
A	color-n35-k4-s013.cnf	U	0.01	0.009	B+C	EDP3-11000	S	28.3	28.9
A	color-n38-k3-s008.cnf	U	0.01	0.009	B+C	FmlaEquivChain_4_6_6.sanitized	U	0.567	0.568
A	over-n100-s016.cnf	U	0.048	0.047	B+C	PRP_30_38	S	18.6	19
A	over-n110-s002.cnf	U	0.035	0.036	B+C	SC25_Timetable_C_392_E_45_Cl_25_D_7_T_50.normalised	S	47.3	48.2
A	over-n120-s004.cnf	U	0.036	0.038	B+C	SCPC-500-5	U	10.2	10.3
A	over-n120-s006.cnf	U	0.038	0.038	B+C	T105.2.0	U	308	311
A	over-n120-s018.cnf	U	0.035	0.036	B+C	asconhashv12_opt64_H4_M2-LOD9_m0_2_U19.c	U	104	105
A	over-n130-s015.cnf	U	0.038	0.038	B+C	b22_1	U	12.7	13.1
A	over-n130-s023.cnf	U	0.037	0.04	B+C	battleship-14-26-sat	S	14.9	14.8
A	over-n150-s021.cnf	U	0.047	0.046	B+C	circuit_32in32out_with_500gates_6in6out_dist64_seed1.sanitized	S	0.868	0.868
A	over-n160-s013.cnf	U	0.05	0.053	B+C	circuit_48in64out_with_800gates_4in4out_dist128_seed4.sanitized	S	9.23	9.38
A	over-n180-s012.cnf	U	0.059	0.059	B+C	clique_n2_k10.sanitized	U	0.117	0.117
A	over-n180-s017.cnf	U	0.05	0.049	B+C	combined-crypto1-wff-seed-1-wffvars-450-cryptocplx-40-overlap-2	S	9.78	9.93
A	over-n210-s022.cnf	U	0.097	0.117	B+C	hwmccl17miters-xits-iso-oski15a08b00s.sanitized	S	3.92	3.93
A	over-n240-s020.cnf	U	0.182	0.187	B+C	mod4block_2vars_10gates_u2_autoenc-sc2009	S	3.47	3.52
A	over-n250-s008.cnf	U	0.244	0.245	B+C	mp1-Nb5T15	S	121	122
A	over-n250-s019.cnf	U	0.169	0.183	B+C	mp1-blockpuzzle_5x10_s7_free4	U	7.18	7.24
A	over-n270-s007.cnf	U	0.321	0.349	B+C	mp1-klieber2017s-0500-023-t12	S	27.6	27.2
A	over-n270-s010.cnf	U	0.165	0.18	B+C	oddball_13_5_ttf.normalised	S	20.9	20.9
A	over-n280-s000.cnf	U	0.436	0.45	B+C	or_randxor_k3_n540_m540	U	19.1	19.3
A	over-n280-s014.cnf	U	0.431	0.444	B+C	pyhala-braun-sat-35-4-04.shuffled	S	2.72	2.72
A	over-n50-s001.cnf	U	0.035	0.025	B+C	qwh.50.1250.shuffled-as.sat03-1655	S	11.2	11.5
A	over-n50-s005.cnf	U	0.01	0.01	B+C	rsbat-v760c43649gyes10	S	13.2	13.2
A	over-n60-s003.cnf	U	0.015	0.016	B+C	square.2.0.i.sm2-cvc4	U	4.68	4.72
A	over-n80-s009.cnf	U	0.028	0.027	B+C	sted5_0x24204-50	S	3.07	3.12
A	over-n80-s011.cnf	U	0.02	0.02	B+C	summle_X4044_steps7_11-2-2-4-4-8-25-100	S	7.78	7.79
A	over-n80-s024.cnf	U	0.03	0.029	B+C	unsat-set-b-fclqcolor-10-07-09.sat05-1282.reshuffled-07	U	0.067	0.067
A	pebble-10.cnf	U	0.031	0.03	B+C	veer_axi_yosyshq_appnote_123_veer_axi-p23_transition	U	8.38	8.29
A	pebble-11.cnf	U	0.041	0.041	B+C	x9-08014.sat.sanitized	U	11.6	11.6
A	pebble-12.cnf	U	0.052	0.066	R	04ded-mdp-28-14-sat	S	0.067	0.066
A	pebble-13.cnf	U	0.07	0.07	R	05c8e-noL-11-2.sanitized	S	289	285
A	pebble-14.cnf	U	0.091	0.095	R	093fa-128_125.sanitized	S	108	115
A	pebble-15.cnf	U	0.122	0.12	R	0a27e-noL-11-0.sanitized	S	296	294
A	pebble-4.cnf	U	0.009	0.009	R	0a488-b19_1	U	502	512
A	pebble-5.cnf	U	0.024	0.03	R	0a84sum_of_three_cubes_42_known_representation	S	76.1	76.7
A	pebble-6.cnf	U	0.012	0.012	R	0ae91-dislog_al4_x14_n24	S	648	648
A	pebble-7.cnf	U	0.013	0.014	R	0d316-linked_list_swap_contents_safety_unwind65	U	76.1	74.6
A	pebble-8.cnf	U	0.026	0.017	R	0f262-linked_list_swap_contents_safety_unwind73	U	58.2	57.7
A	pebble-9.cnf	U	0.023	0.022	R	113e6-oisc-subrv-sll-nested-8	U	234	236
A	php-5.cnf	U	0.008	0.008	R	15456-hwmc20miters-iso-rast-p06.sanitized	U	1.42	1.42
A	php-6.cnf	U	0.008	0.008	R	15456-clique_n5_k10.sanitized	U	1.27	1.27
A	php-7.cnf	U	0.008	0.008	R	16564-ER_400_20_7_apx_2_DS-ST	S	343	335
A	php-8.cnf	U	0.008	0.008	R	180e6-baseballcover11with22_and2positions	S	16.8	17.2
A	tr-n100-s047.cnf	U	0.031	0.049	R	1a20d-hwmc20miters-iso-rast-p11.sanitized	U	1.42	1.42
A	tr-n110-s028.cnf	U	0.035	0.036	R	1a951-linked_list_swap_contents_safety_unwind50	U	53.4	53.7
A	tr-n110-s030.cnf	S	0.028	0.028	R	1c21a-9.normalised	S	59.5	64.5
A	tr-n110-s031.cnf	U	0.033	0.045	R	210fe-32_325.sanitized	S	78.5	82
A	tr-n110-s044.cnf	U	0.031	0.048	R	266e8-mexam_15_12_2.sanitized	U	0.317	0.317
A	tr-n110-s059.cnf	U	0.044	0.044	R	26b64-128_75.sanitized	S	64.9	68
A	tr-n110-s071.cnf	U	0.03	0.034	R	28eeb-oski15a01b40s_opt	U	452	463
A	tr-n110-s072.cnf	S	0.027	0.027	R	29b57-b18	S	236	238
A	tr-n120-s029.cnf	S	0.03	0.029	R	35748-oski15a01b39s_opt	U	399	377
A	tr-n120-s058.cnf	S	0.033	0.05	R	3b5d7-nla-digbench-scaling_dijkstra-u_valuebound1_transition	U	0.918	0.968
A	tr-n120-s095.cnf	S	0.041	0.029	R	4551c-knight_18	S	136	137
A	tr-n130-s024.cnf	S	0.032	0.033	R	5220b-coreMQTT:MQTT_ProcessLoop.dimacs	U	144	144
A	tr-n130-s039.cnf	S	0.031	0.053	R	5314f-oski15a01b15s_opt	U	461	475
A	tr-n130-s040.cnf	S	0.038	0.041	R	59116-oski15a01b06s_opt	U	667	691
A	tr-n140-s004.cnf	U	0.056	0.058	R	59fc7-jiraldezlevy.2200.9086.08.40.149-sr2015	S	141	141
A	tr-n140-s007.cnf	U	0.066	0.064	R	5c478-oski15a01b20s_opt	U	566	578
A	tr-n140-s010.cnf	S	0.05	0.062	R	5ceb9-pj2009_k80	S	130	133
A	tr-n140-s025.cnf	S	0.032	0.033	R	753ca-oski15a01b01s_opt	U	331	341
A	tr-n140-s034.cnf	U	0.084	0.099	R	780b0-rbat-v1375c111739gyes10	S	34.9	35
A	tr-n140-s045.cnf	U	0.052	0.054	R	7a5e9-cfi-rigid-t2-0048-04-or_3_shuffle_all	S	53.4	54
A	tr-n140-s060.cnf	U	0.052	0.052	R	84c6d-128_100.sanitized	S	86.8	90.6
A	tr-n140-s084.cnf	U	0.042	0.043	R	8d4b5-hwmccl17miters-xits-iso-6s281b35.sanitized	U	12	12.2
A	tr-n150-s009.cnf	U	0.046	0.047	R	8ffd7-mdp-32-10-sat	S	0.117	0.117
A	tr-n150-s018.cnf	U	0.098	0.12	R	912ce-hwmc20miters-iso-mul2.sanitized	U	0.717	0.768
A	tr-n150-s062.cnf	U	0.092	0.097	R	9376a-cabp-X-can_715.mtx.rnd-k102.sanitized	S	125	126
A	tr-n150-s078.cnf	U	0.081	0.095	R	9541d-cabp-V-nos6.mtx.rnd-k275.sanitized	S	224	225
A	tr-n150-s087.cnf	U	0.097	0.118	R	99478-noL-11-6.sanitized	S	325	325
A	tr-n160-s008.cnf	U	0.122	0.13	R	9aa85-schoolt-5-3-8-2-4-5-10-0-1-0-2-seed4.sanitized	U	190	190
A	tr-n160-s054.cnf	S	0.054	0.077	R	9cd3a-FmlaEquivChain_4_6_6.sanitized	U	0.517	0.567
A	tr-n160-s056.cnf	U	0.122	0.145	R	a14d9-64_200.sanitized	S	85.5	89.2
A	tr-n160-s089.cnf	U	0.14	0.143	R	a49ec-grs-32-128	U	470	473
A	tr-n160-s097.cnf	U	0.131	0.136	R	ab02c-sum_of_three_cubes_906_known_representation	S	17.2	17.4
A	tr-n170-s033.cnf	S	0.1	0.103	R	ab17e-vdwb_k6_n250.sanitized	S	123	119
A	tr-n170-s053.cnf	U	0.186	0.204	R	ac388-pj2008_k80	S	141	144
A	tr-n170-s068.cnf	U	0.137	0.166	R	efc1b-oddball_19_4_ttf.normalised	U	36	35.8
A	tr-n180-s020.cnf	U	0.096	0.116	R	fb0e5-constraints_17_0_4_2.sanitized	U	43.6	43.8
A	tr-n180-s036.cnf	U	0.14	0.145	W	043c9-miles1500.72.sanitized	U	3.97	3.97
A	tr-n180-s037.cnf	S	0.064	0.065	W	08d56-cliquecoloring_n26_k7_c6	U	0.117	0.116
A	tr-n180-s090.cnf	S	0.067	0.074	W	0c043-oddball_53_5_tto_zp.normalised	S	5.53	5.48
A	tr-n190-s015.cnf	S	0.061	0.081	W	0d082-mexam_23_20_2.sanitized	U	3.17	3.17
A	tr-n190-s023.cnf	U	0.173	0.194	W	0ec6b-count_p3_M19.sanitized	U	0.968	0.968
A	tr-n190-s048.cnf	S	0.042	0.043	W	0f107-mulsol1.2.30.sanitized	U	0.667	0.668
A	tr-n190-s050.cnf	S	0.059	0.057	W	12820-RoundRobin_n16_d14	U	0.066	0.066
A	tr-n190-s081.cnf	U	0.226	0.235	W	12ac1-exam_75_65.sanitized	U	0.517	0.467
A	tr-n200-s000.cnf	U	0.327	0.31	W	16c54-two-trees-511v.sanitized	S	0.034	0.034
A	tr-n200-s049.cnf	U	0.331	0.341	W	16f8b-zeroin.i.3.29.sanitized	U	3.57	3.58
A	tr-n210-s096.cnf	U	0.484	0.515	W	18c0e-cliquecolouring_n31_k5_c4.sanitized	U	0.117	0.117
A	tr-n220-s013.cnf	S	0.43	0.446	W	19951-oddball_80_5_tto_zp.normalised	S	12.3	12.1
A	tr-n220-s035.cnf	U	0.56	0.576	W	1a3be-le450_25b.col.25	U	1.77	1.77

Tier	Benchmark	S/U	C++	C	Tier	Benchmark	S/U	C++	C
A	tr-n220-s076.cnf	U	0.439	0.444	W	1fb64-chnl-040x041_sanitized	U	0.417	0.417
A	tr-n220-s092.cnf	S	0.08	0.073	W	20273-chnl-050x051_sanitized	U	1.02	1.02
A	tr-n230-s042.cnf	U	1.1	1.12	W	247e4-16_sanitized	S	0.217	0.217
A	tr-n230-s063.cnf	S	0.064	0.065	W	2563e-1x1x13_1x3x6_pair_2_1_1_4_2_0_sanitized	TO	1200	1200
A	tr-n240-s012.cnf	S	0.193	0.199	W	29e61-worker_30_60_25_0_9	U	0.768	0.769
A	tr-n240-s014.cnf	U	1.44	1.48	W	2b60f-sgen1-unsat-121-100	U	0.067	0.066
A	tr-n240-s026.cnf	S	0.064	0.066	W	2c3c2-mchess_17	U	12.4	12.5
A	tr-n240-s051.cnf	U	1.94	1.98	W	2e458-oddball_70_5_tto_zp.normalised	S	9.88	9.93
A	tr-n240-s073.cnf	S	0.072	0.07	W	2d89f-2_sanitized	S	0.217	0.217
A	tr-n250-s001.cnf	S	0.113	0.115	W	306a2-lattice-25_24_sanitized	U	3.32	3.32
A	tr-n250-s043.cnf	U	3.82	3.89	W	30c6f-6_sanitized	S	0.217	0.217
A	tr-n250-s067.cnf	S	0.068	0.067	W	35e0b-count_p2_M33_sanitized	U	0.467	0.467
A	tr-n250-s079.cnf	U	4.02	4.09	W	35ec9-MVRoundRobin_n14_d10_v2	U	1.72	1.72
A	tr-n250-s099.cnf	U	4.03	4.13	W	3a1f1-MVRoundRobin_n16_d10_v3	U	5.78	5.73
A	tr-n260-s002.cnf	U	5.69	5.78	W	3a9df-ntl-90d-40_sanitized	S	93.7	90.3
A	tr-n260-s085.cnf	U	5.8	5.83	W	41d14-tseitin_d3_n100000	U	0.617	0.618
A	tr-n270-s016.cnf	U	3.81	3.87	W	43d40-php_sudoku_p15_h14_sanitized	U	0.034	0.034
A	tr-n270-s021.cnf	S	2.36	2.38	W	44c25-Break_20_72.xml	TO	1200	1200
A	tr-n270-s027.cnf	S	0.069	0.071	W	45742-mchess_20	U	17.6	17.8
A	tr-n280-s052.cnf	U	13.8	13.8	W	45a09-fermat-931960058139995587	S	0.066	0.067
A	tr-n280-s057.cnf	U	6.94	7.01	W	4d84e-4_sanitized	S	0.217	0.217
A	tr-n280-s075.cnf	S	2.37	2.42	W	511f1-14_sanitized	S	0.217	0.217
A	tr-n280-s088.cnf	S	0.716	0.76	W	53e34-RoundRobin_n17_d15	U	0.117	0.117
A	tr-n290-s038.cnf	S	0.932	0.957	W	54d49-hwmcc17miters-xits-iso-6s163_sanitized	U	0.417	0.417
A	tr-n290-s080.cnf	U	5.32	5.36	W	58eaf-count_p2_M41_sanitized	U	1.17	1.17
A	tr-n290-s086.cnf	S	2.38	2.42	W	5d8b4-mexam_25_20_2_sanitized	U	4.07	4.12
A	tr-n300-s041.cnf	S	0.301	0.33	W	626c3-snw_16_9_CCSpre	TO	1200	1200
A	tr-n300-s055.cnf	U	23.1	23.4	W	65bf8-DSC125_9.30_sanitized	U	0.517	0.517
A	tr-n300-s093.cnf	S	0.845	0.877	W	67c53-cliquecolouring_n15_k9_c8_sanitized	U	0.117	0.117
A	tr-n50-s022.cnf	S	0.011	0.011	W	695e2-rbsat-v1150c84314gyes1	S	211	210
A	tr-n50-s061.cnf	S	0.011	0.009	W	6ddda-inithx.i.2.30_sanitized	U	6.48	6.53
A	tr-n50-s066.cnf	U	0.03	0.022	W	6f6a2-RoundRobin_n18_d15	U	0.117	0.116
A	tr-n50-s077.cnf	U	0.014	0.028	W	71849-exam_85_70_sanitized	U	0.717	0.718
A	tr-n60-s046.cnf	U	0.023	0.036	W	727eb-count_p2_M25_sanitized	U	0.167	0.167
A	tr-n60-s064.cnf	S	0.02	0.02	W	73721-DSC125_9.25_sanitized	U	0.417	0.417
A	tr-n60-s074.cnf	U	0.02	0.022	W	74423-10_sanitized	S	0.217	0.217
A	tr-n60-s082.cnf	S	0.014	0.034	W	757d6-chnl-030x091_sanitized	U	0.868	0.868
A	tr-n60-s098.cnf	S	0.027	0.026	W	76895-cliquecolouring_n13_k8_c7_sanitized	U	0.067	0.066
A	tr-n70-s005.cnf	U	0.026	0.041	W	78e8f-RoundRobin_n17_d14	U	0.116	0.117
A	tr-n70-s032.cnf	S	0.03	0.031	W	7b962-exam_90_80_sanitized	U	1.02	0.968
A	tr-n80-s006.cnf	S	0.009	0.009	W	7db30-two-trees-1023v_sanitized	S	0.066	0.066
A	tr-n80-s017.cnf	U	0.008	0.009	W	7e58b-mp1_Nb7T45	S	1.77	1.82
A	tr-n80-s065.cnf	S	0.03	0.031	W	86078-mexam_17_15_2_sanitized	U	0.718	0.718
A	tr-n80-s083.cnf	S	0.027	0.042	W	87378-hwmcc17miters-xits-iso-6s299b685_sanitized	U	1.47	1.52
A	tr-n80-s091.cnf	U	0.039	0.032	W	8e1cd-15_sanitized	S	0.217	0.217
A	tr-n90-s003.cnf	S	0.039	0.038	W	8fcde-chnl-020x061_sanitized	U	0.217	0.217
A	tr-n90-s011.cnf	S	0.028	0.048	W	9078e-13_sanitized	S	0.217	0.217
A	tr-n90-s019.cnf	U	0.035	0.052	W	90843-MVRoundRobin_n20_d10_v2	U	6.33	6.33
A	tr-n90-s069.cnf	S	0.023	0.023	W	92b22-exam_90_70_sanitized	U	0.818	0.818
A	tr-n90-s070.cnf	S	0.031	0.03	W	981c4-tseitin_grid_n250_m250	U	0.868	0.868
A	tr-n90-s094.cnf	U	0.033	0.037	W	99749-mulsol.i.4.30_sanitized	U	0.667	0.667
A	under-n100-s004.cnf	S	0.022	0.011	W	9ba81-latin_square_10.15_sanitized	U	8.08	8.18
A	under-n100-s007.cnf	S	0.025	0.033	W	9d7ca-ncc_none_21015_5_3_3_0_0_11	U	169	171
A	under-n100-s022.cnf	S	0.023	0.019	W	a09eb-RoundRobin_n18_d16	U	0.117	0.117
A	under-n110-s016.cnf	S	0.017	0.018	W	a5dc6-mdp-36-10-sat	S	0.467	0.468
A	under-n120-s017.cnf	S	0.01	0.03	W	a7e5d-mexam_20_18_2_sanitized	U	1.67	1.72
A	under-n140-s023.cnf	S	0.033	0.024	W	a8608-rbsat-v945c61409g3	S	108	108
A	under-n170-s019.cnf	S	0.025	0.028	W	a95f0-chnl-030x061_sanitized	U	0.417	0.418
A	under-n180-s006.cnf	S	0.01	0.011	W	aa4e8-exam_75_70_sanitized	U	0.567	0.567
A	under-n210-s002.cnf	S	0.011	0.011	W	ab2ab-1_sanitized	S	0.217	0.217
A	under-n210-s024.cnf	S	0.011	0.01	W	ab343-mdp-36-14-sat	S	0.467	0.467
A	under-n240-s011.cnf	S	0.011	0.012	W	ac347-cliquecolouring_n41_k5_c4_sanitized	U	1.02	1.02
A	under-n250-s015.cnf	S	0.012	0.012	W	aca9f-12_sanitized	S	0.217	0.217
A	under-n260-s013.cnf	S	0.037	0.037	W	b0142-count_p2_M21_sanitized	U	0.117	0.117
A	under-n270-s001.cnf	S	0.038	0.054	W	b1c8e-cliquecolouring_n15_k7_c6_sanitized	U	0.066	0.066
A	under-n290-s018.cnf	S	0.042	0.042	W	b358b-crm_40_1521_s	S	0.066	0.067
A	under-n290-s020.cnf	S	0.041	0.041	W	b3c58-StConn_7_128_sanitized	U	9.13	9.18
A	under-n300-s014.cnf	S	0.041	0.042	W	bffe2-cliquecolouring_n21_k6_c5_sanitized	U	0.066	0.066
A	under-n50-s000.cnf	S	0.008	0.024	W	bcdc8-chnl-020x101_sanitized	U	0.467	0.468
A	under-n50-s003.cnf	S	0.017	0.008	W	bea83-xor_op_n36_d3	U	75.4	75.9
A	under-n50-s009.cnf	S	0.008	0.024	W	c05d4-MVRoundRobin_n16_d10_v2	U	2.67	2.67
A	under-n80-s005.cnf	S	0.009	0.009	W	c4476-mexam_14_12_2_sanitized	U	0.267	0.267
A	under-n80-s010.cnf	S	0.017	0.022	W	cb662-RoundRobin_n17_d13	U	0.117	0.117
A	under-n90-s008.cnf	S	0.015	0.015	W	d1194-mdp-32-14-sat	S	0.317	0.317
A	under-n90-s012.cnf	S	0.01	0.026	W	d1335-mulsol.i.1.48_sanitized	U	1.82	1.82
A	under-n90-s021.cnf	S	0.008	0.01	W	d27f7-exam_95_70_sanitized	U	0.918	0.868
A	xor-n100-s004.cnf	S	0.008	0.008	W	d35df-16_2	S	28.2	28.2
A	xor-n110-s017.cnf	U	0.006	0.008	W	db15a-mdp-32-11-sat	S	0.117	0.116
A	xor-n139-s003.cnf	U	0.007	0.008	W	ddf96-oddball_57_5_tto_zp.normalised	S	6.32	6.33
A	xor-n140-s006.cnf	S	0.008	0.008	W	e3a13-9_sanitized	S	0.167	0.167
A	xor-n150-s002.cnf	S	0.009	0.009	W	e5787-fpsol2.i.2.29_sanitized	U	2.32	2.32
A	xor-n150-s013.cnf	U	0.008	0.007	W	e9125-lattice-20.19_sanitized	U	0.818	0.818
A	xor-n157-s012.cnf	S	0.009	0.008	W	f3abc-tseitin_grid_n400_m400	U	2.12	2.12
A	xor-n158-s009.cnf	U	0.01	0.008	W	f8342-php_sudoku_p20_h19_sanitized	U	0.034	0.034
A	xor-n184-s010.cnf	S	0.009	0.009	W	fa50d-circuit_32in64out_with_150gates_6in6out_dist256_seed1_sanitized	S	1.42	1.47
A	xor-n189-s018.cnf	S	0.009	0.011	W	fbf99-count_p2_M45_sanitized	U	1.82	1.82
A	xor-n195-s015.cnf	U	0.008	0.016	W	fed3f-8_sanitized	S	0.217	0.217
A	xor-n200-s014.cnf	S	0.01	0.01					

C.3 Case studies

The seed identifies each of the ten *cruxmiter* instances ($N = 32$); the node identifies each of the 23 Chvátal subproblems (finite case 7). All runs returned UNSAT with accepted certificates. Each cell lists solving time / checking time / proof size (GB).

Table 10. Arithmetic verification.

Seed	TACTIX	TACTIX _{mi}	KISSAT 4.0.4
0	2170/8070/4.64	823/1510/1.89	883/1830/1.96
1	2170/8220/4.64	827/1520/1.89	881/1830/1.96
2	276/9770/6.98	224/432/0.54	599/1280/1.53
3	1810/5560/3.61	277/487/0.67	342/662/0.802
4	4370/13300/7.97	1390/3170/3.3	6100/12500/11.6
5	587/2050/1.47	122/194/0.311	741/1600/1.71
6	438/1480/1.05	284/491/0.703	460/915/1.14
7	197/5990/4.99	41.3/64.1/0.102	116/198/0.324
8	597/2060/1.44	454/963/1.28	819/1870/1.97
9	2870/9530/5.5	422/803/0.931	5080/10500/9.25

Table 11. Chvátal's conjecture.

Node	TACTIX _{ch}	TACTIX SATSUMA+KISSAT	KISSAT 4.0.4
11111001	525/1800/0.759	1600/3760/2.09	1430/5880/1.97
111110011	528/1820/0.759	1630/3640/2.09	1620/2770/2.15
1111100110	344/11600/4.77	891/2090/1.17	1110/4220/1.49
11111001100	340/1120/0.477	869/1970/1.17	1100/4030/1.49
1111110010000	511/1790/0.698	1210/2500/1.6	1630/6110/2.07
11111100100001	515/1790/0.698	1130/2390/1.52	1790/6470/2.13
11111100100010	515/1810/0.698	1120/2230/1.52	1670/6320/2.13
111111001000100	253/691/0.304	411/903/0.575	418/1660/0.643
1111111001001	261/694/0.305	547/1180/0.797	630/2320/0.852
1111110010010	246/674/0.298	419/951/0.603	550/2150/0.788
1111110010100	183/425/0.197	305/730/0.443	486/2000/0.718
1111110010101	445/1420/0.599	1100/2430/1.49	1160/4570/1.62
11111100101010	317/937/0.399	670/1430/0.939	839/3170/1.09
111111001010101	315/947/0.399	772/1730/1.02	904/3390/1.22
1111110010101011	314/941/0.399	732/1620/1.03	834/3190/1.16
11111100101010110	313/937/0.399	729/1630/1.03	824/3190/1.16
1111110011	573/1940/0.821	1320/2970/1.79	1770/6740/2.23
11111100110	300/838/0.381	578/1270/0.778	593/2360/0.793
111111001100	296/838/0.381	572/1270/0.778	570/2260/0.793
1111110011001	299/801/0.377	440/949/0.623	473/1880/0.69
11111100110011	298/796/0.377	424/917/0.59	489/1890/0.692
11111110100	525/1680/0.712	955/2050/1.24	997/3770/1.32
111111101000	247/696/0.326	375/752/0.484	406/1550/0.541