CircuitTSAT

A Solver for Large Instances of the Disjunctive Temporal Problem

Blaine Nelson[1] T. K. Satish Kumar[2]

[1]nelsonb@eecs.berkeley.edu
Department of Computer Science
University of California at Berkeley

[2]skumar@ihmc.us
Institute for Human and Machine Cognition
Pensacola, FL, USA

Introduction

- We study fast solution to the Disjunctive Temporal Problem (DTP)
- We present CircuitTSAT that uses circuit-based approaches similar to [Bryant et al, 2007] for DTPs.
  - We show that structure of the DTP is amenable to circuit representations
- We explore how our approach compares with other approaches to other DTP solvers and discuss the advantages and disadvantages of CircuitTSAT

Motivating Example

Scheduling

- Consider set of N events $\{X_i\}$
- The i-th event has duration d_i
- Some events *must* occur before others:

$$X_i \prec X_j$$

- Single Resources: force events to be disjoint

$$X_i \prec X_j \ \vee \ X_j \prec X_i$$

Motivating Example

Scheduling

- Consider set of N events $\{X_i\}$
- The i-th event has duration d_i
- Some events *must* occur before others:

$$X_j - X_i \geq d_i$$

A *simple temporal constraint*!

- Single Resources: force events to be disjoint

$$X_i \prec X_j \ \vee \ X_j \prec X_i$$

Motivating Example

Scheduling

- Consider set of N events $\{X_i\}$
- The i-th event has duration d_i
- Some events *must* occur before others:

$$X_j - X_i \geq d_i$$

A *simple temporal constraint*!

- Single Resources: force events to be disjoint

$$X_j - X_i \geq d_i \quad \vee \quad X_i - X_j \geq d_j$$

A disjunction of simple temporal constraints

Simple Temporal Problems

- An STP is characterized by
  1. A set of events: $\mathcal{X} = \{X_0, X_1 \ldots X_N\}$
  2. A set of simple temporal constraints (STCs) of the form

$$LB(i,j) \leq X_j - X_i \leq UB(i,j)$$

- A solution is an assignment $\mathcal{X}$ satisfying all STCs

- STPs can be efficiently solved using shortest path algorithms [Dechter et al, 1991]

Disjunctive Temporal Problems

- A DTP is characterized by
  1. Events: $\mathcal{X} = \{X_0, X_1 \ldots X_N\}$
  2. Disjunctive temporal constraints (DTCs) of the form

$$\mathcal{S}_1(z_1) \vee \mathcal{S}_2(z_2) \vee \ldots \vee \mathcal{S}_K(z_K)$$

  where $\mathcal{S}_k(z_k)$ is an STC of the form

$$X_{i_k} - X_{j_k} \leq z_k$$

  for a pair of temporal variables
- A solution is an assignment $\mathcal{X}$ satisfies every DTCs.
- Solving DTPs are NP-hard in general.

Previous Work

- **TSAT++** [Armando et al, 2000, Armando et al, 2004]
  - Each STC is represented by a propositional variable
  - Satisfying the resulting SAT instance yields an STP
  - Consistency of the induced STP finds DTP solutions
- **Satisfiability Modulo Theories (SMT) Solvers**
  - SMT solvers generalize SAT instances
  - Domain-specific functions replace Boolean variables
  - DTPs can be cast as SMT(DL)
- **UCLID** [Bryant et al, 2002, Bryant et al, 2007]
  - UCLID SMT solver employs circuit-based expansions
  - Modern SMT solvers outperform circuit-based approaches
  - We re-explore circuit-based approaches for DTPs

CircuitTSAT

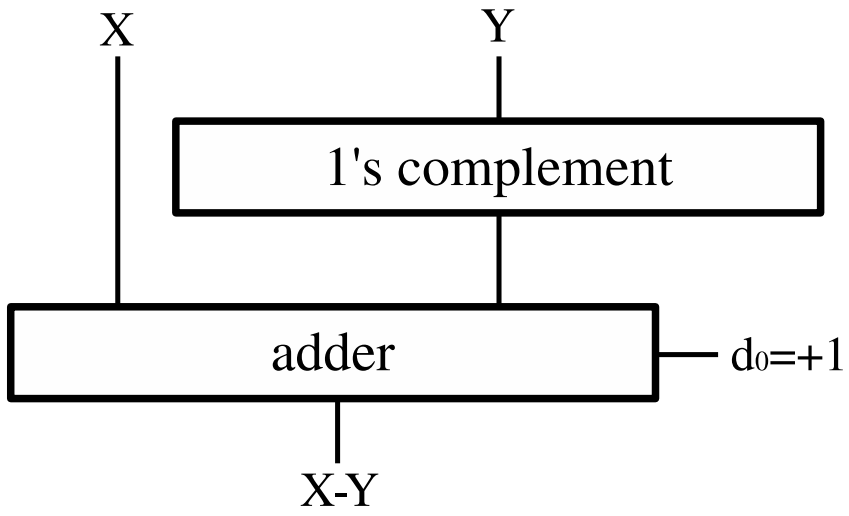
A circuit-based approach to DTPs

Representing STCs as Circuits

- Let X and Y be integral variables.
- Further, suppose both are between 0 and Q.
- Then both can represented by $q = \lceil \log_2 Q \rceil$ bits:

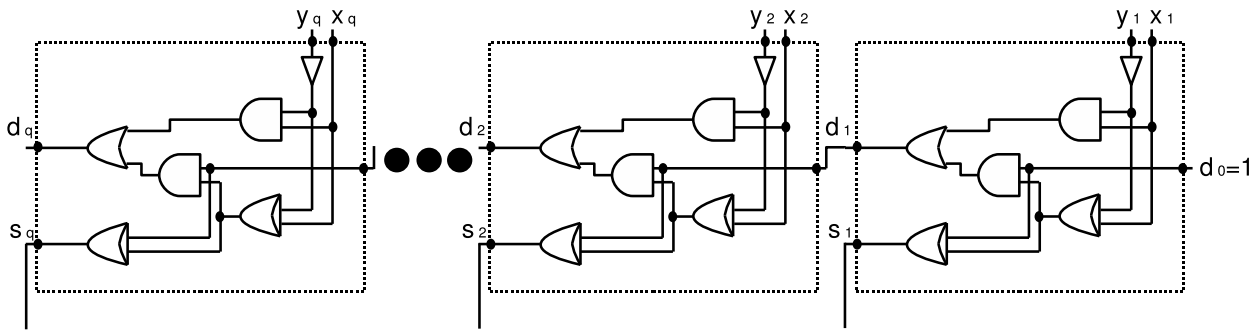
$$\begin{aligned} X &= \langle x_q, x_{q-1} \ldots x_1 \rangle \\ Y &= \langle y_q, y_{q-1} \ldots y_1 \rangle \end{aligned}$$

- $X - Y$ can then be represented as a circuit:



Representing STCs as Circuits

Sum and Carry Bits



- The $X - Y$ difference circuit produces sum S and carry D bits:

$$\begin{aligned} S &= \langle s_q, s_{q-1} \ldots s_1 \rangle \\ D &= \langle d_q, d_{q-1} \ldots d_1, d_0 = 1 \rangle \end{aligned}$$

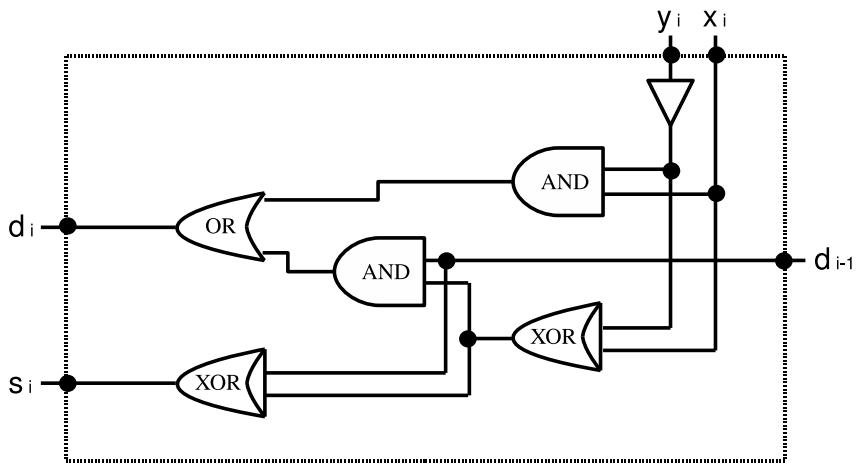
- Since $X \geq 0$ and $Y \geq 0$, $X - Y$ cannot overflow.
- Further, d_q indicates the sign of $X - Y$:

$$d_q \quad \Leftrightarrow \quad X - Y \geq 0$$

Representing STCs as Circuits

Auxiliary Variables for $X - Y \geq 0$

Consider a Full Adder Circuit



We construct auxiliary variables d_i

$$\begin{aligned} d_i &\Leftrightarrow [(x_i \oplus \bar{y}_i) \wedge d_{i-1}] \vee (x_i \wedge \bar{y}_i) \\ &\Leftrightarrow [(x_i \vee \bar{y}_i) \wedge d_{i-1}] \vee (x_i \wedge \bar{y}_i) \end{aligned}$$

Representing STCs as Circuits

Extending to $Y - X \leq z$

- Constraints of $X - Y \geq 0$ extend to $Y - X \leq z$.
- By regrouping, we convert it to 2 sums:

$$(X - Y) + z \geq 0$$

Representing STCs as Circuits

Extending to $Y - X \leq z$

- Constraints of $X - Y \geq 0$ extend to $Y - X \leq z$.
- By regrouping, we convert it to 2 sums:

$$\underbrace{(X - Y)}_{1^{st}\text{ term} = S} \quad + z \geq 0$$

- The 1^{st} is the difference between two variables: S.

Representing STCs as Circuits

Extending to $Y - X \leq z$

- Constraints of $X - Y \geq 0$ extend to $Y - X \leq z$.
- By regrouping, we convert it to 2 sums:

$$\underbrace{S + z}_{2^{\text{nd}}\ \text{term}} \geq 0$$

- The 2^{nd} adds a constant to that difference.

Representing STCs as Circuits

Extending to $Y - X \leq z$

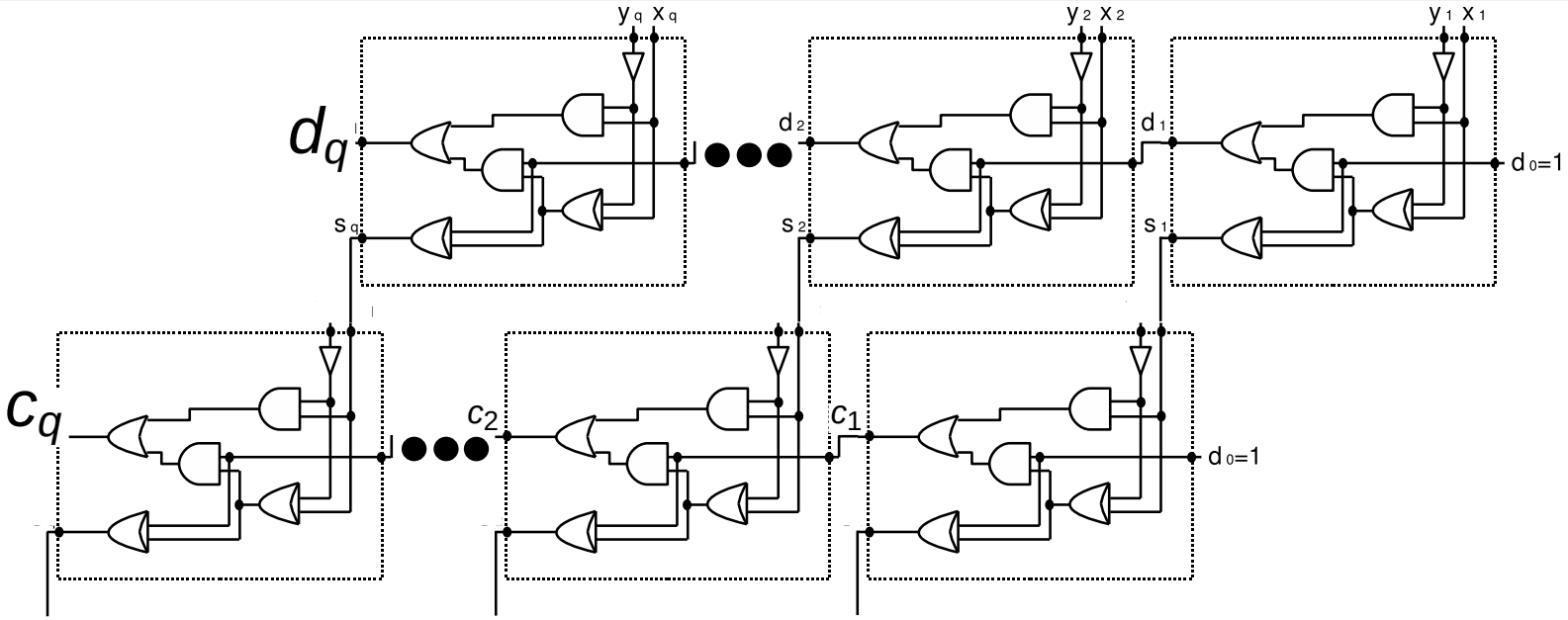
- Constraints of $X - Y \geq 0$ extend to $Y - X \leq z$.
- By regrouping, we convert it to 2 sums:

$$S + z \geq 0$$

- We need a circuit for $S + z \geq 0$

Representing STCs as Circuits

Extending to $S + z \geq 0$



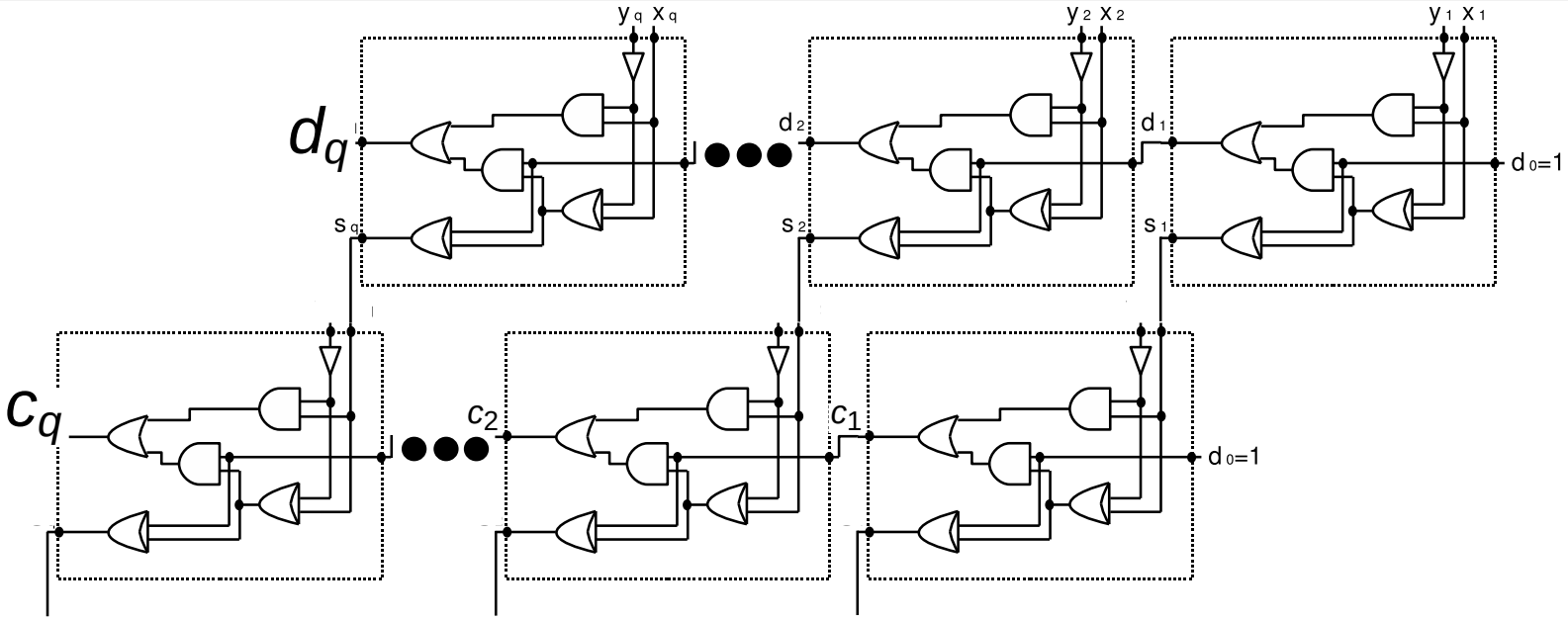
- Terms of S are $s_i \Leftrightarrow d_{i-1} \oplus (x_i \oplus \bar{y}_i)$
- Carry bits of $S + z$ compiled from bits of z

$$c_i \Leftrightarrow [(s_i \vee \textcircled{z}_i) \wedge c_{i-1}] \vee (s_i \wedge \textcircled{z}_i)$$

where $\textcircled{z}_i$ depends on sign of z

Representing STCs as Circuits

Extending to $S + z \geq 0$



- Terms of S are $s_i \Leftrightarrow d_{i-1} \oplus (x_i \oplus \bar{y}_i)$
- Carry bits of $S + z$ compiled from bits of z

$$c_i \Leftrightarrow \begin{cases} c_{i-1} \vee s_i & \text{if } \textcircled{z}_i = 1 \\ c_{i-1} \wedge s_i & \text{if } \textcircled{z}_i = 0 \end{cases}$$

where $\textcircled{z}_i$ depends on sign of z

Representing STCs as Circuits

Extending to $S + z \geq 0$

- $X - Y + z \geq 0$ expressed by c_q but it can overflow
- There are 2 cases:
  1. $X - Y$ and z agree in sign
     - Adding z to $X - Y$ will not change its sign
     - Hence d_q indicates the sign of $X - Y + z$
  2. $X - Y$ and z have opposite sign
     - Adding z to $X - Y$ cannot overflow
     - Hence c_q indicates the sign of $X - Y + z$
- We can represent the DTCs state by
  - $d_q \wedge c_q$ if $z_w \leq 0$
  - $d_q \vee c_q$ if $z_w > 0$

Representing DTCs Propositionally

- DTCs are disjunctions of STCs.
- The w^{th} STC $Y - X \leq z$ can be represented propositionally by a_w:

$$\begin{aligned} a_w &\Leftrightarrow d_{q,w} \wedge c_{q,w} \quad \text{if } z_w \leq 0 \\ a_w &\Leftrightarrow d_{q,w} \vee c_{q,w} \quad \text{if } z_w > 0 \end{aligned}$$

- The DTC can then be expressed as a CNF clause:

$$a_1 \vee a_2 \vee \ldots \vee a_k$$

CircuitTSAT: Representing DTPs as CNF

- A DTP is expressed in CNF by difference logic:
  - Carry bit d_q of $X - Y$ is expressed in CNF
  - Carry bit c_q of $X - Y + z$ is expressed in CNF
  - w^{th} STC $Y - X \leq z$ is expressed in CNF in terms of $c_{q,w}$ and $d_{q,w}$ as a_w.
  - Each DTC is represented in CNF

- Solutions to the DTP are found using modern SAT Solver. We tested several solvers:
  - JeruSAT [Nadel, 2002]
  - MiniSAT [Eén and Sörensson, 2003]
  - zChaff

Trade-offs with CircuitTSAT

- Size of CNF (For a DTP with N temporal variables, M clauses, and K disjuncts per clause)
  - $O(qN + qKM)$ propositional variables
  - $O(qKM)$ clauses
- Choosing the size of the bit-space
  - Worst case: Chain of largest constant from each DTC
  - In a chain, these constants sum

$$q_{max} = \left\lceil \log_2 \sum_{i=1}^{M} \max_{z_w \in DTC_i} |z_w| \right\rceil$$

  - Smaller q gives faster performance, but is incomplete
- Non-integer problems
  - Negative variables are avoided by translation via X_0
  - Real-valued variables approximated via scaling

Experiments

Experimental Setup

- We compare DTP solvers on random DTPs
- Experimental parameters
  - N — number of temporal variables
  - M/N — ratio of clauses to temporal variables
  - K — number of disjuncts per DTC
  - L — maximum magnitude of any constant in the DTP
- Algorithms we compare
  - CircuitTSAT
  - TSAT++ [Armando et al, 2000, Armando et al, 2004]
  - Yices [Dutertre and de Moura, 2006]

Experiment 1

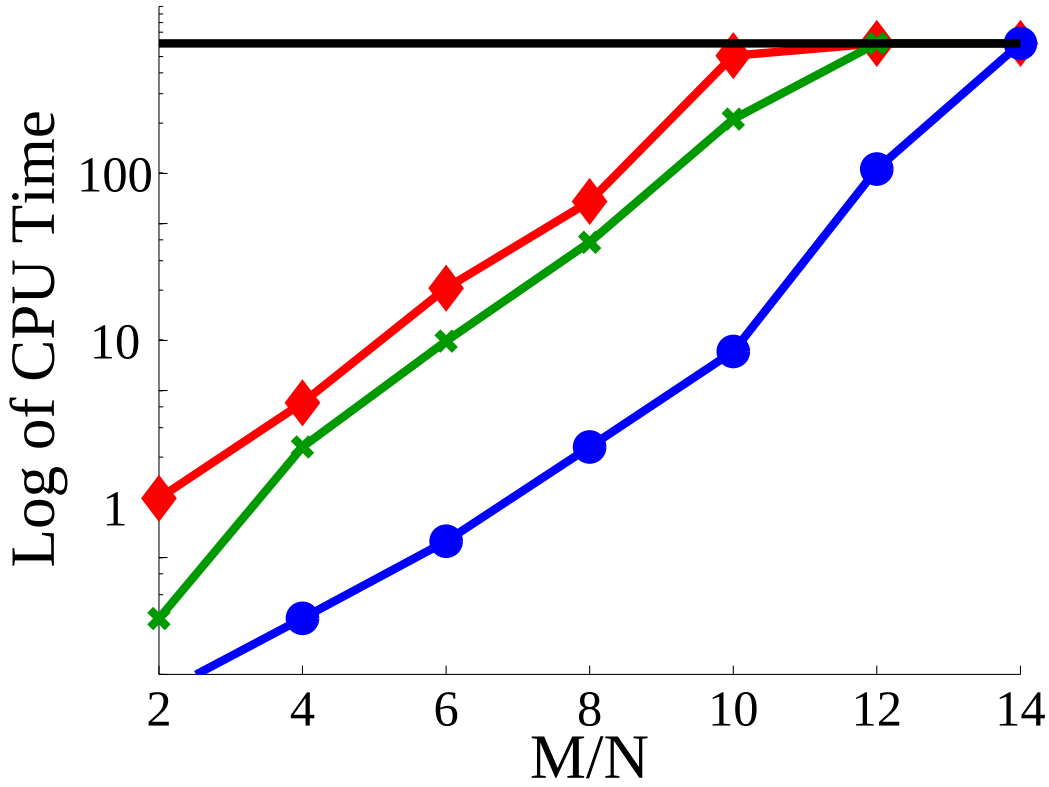
Comparing Performance for an Increasing Number of Literals

- We test for large numbers of temporal variables (N) and many disjuncts per clause (K)
- We plot log exec time vs. clause to variable ratio (M/N)

Performance Comparison of CircuitTSAT

Increasing Number of Literals

Logarithm of Median Running Times for
$K = 3$ and $N = 50$



CircuitTSAT (◆)

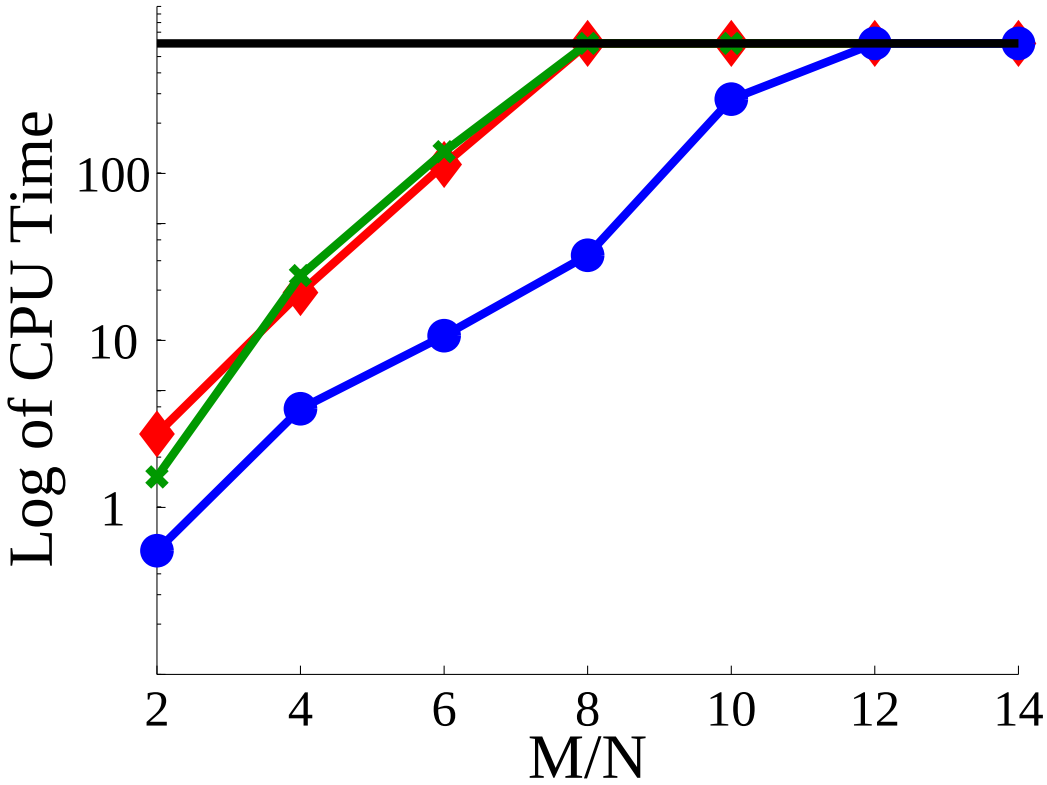
TSAT++ (×)

Yices (●)

Performance Comparison of CircuitTSAT

Increasing Number of Literals

Logarithm of Median Running Times for
$K = 3$ and $N = 100$



CircuitTSAT (◆)

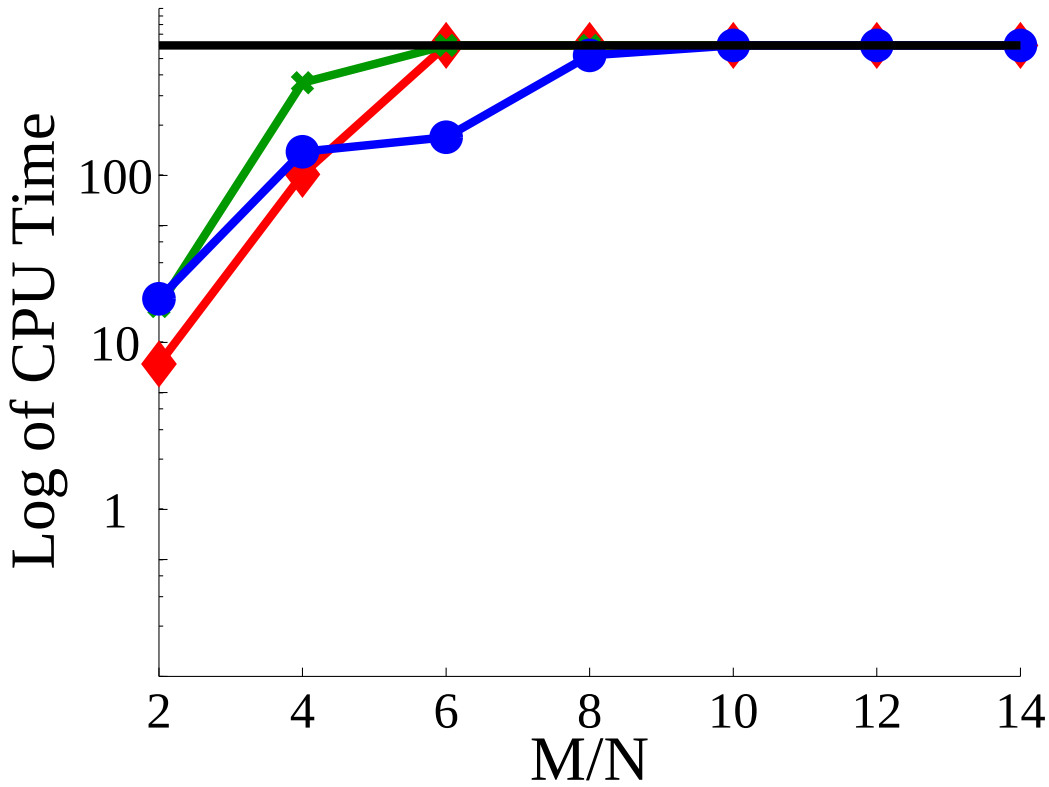
TSAT++ (×)

Yices (●)

Performance Comparison of CircuitTSAT

Increasing Number of Literals

Logarithm of Median Running Times for
$K = 3$ and $N = 200$



CircuitTSAT (◆)

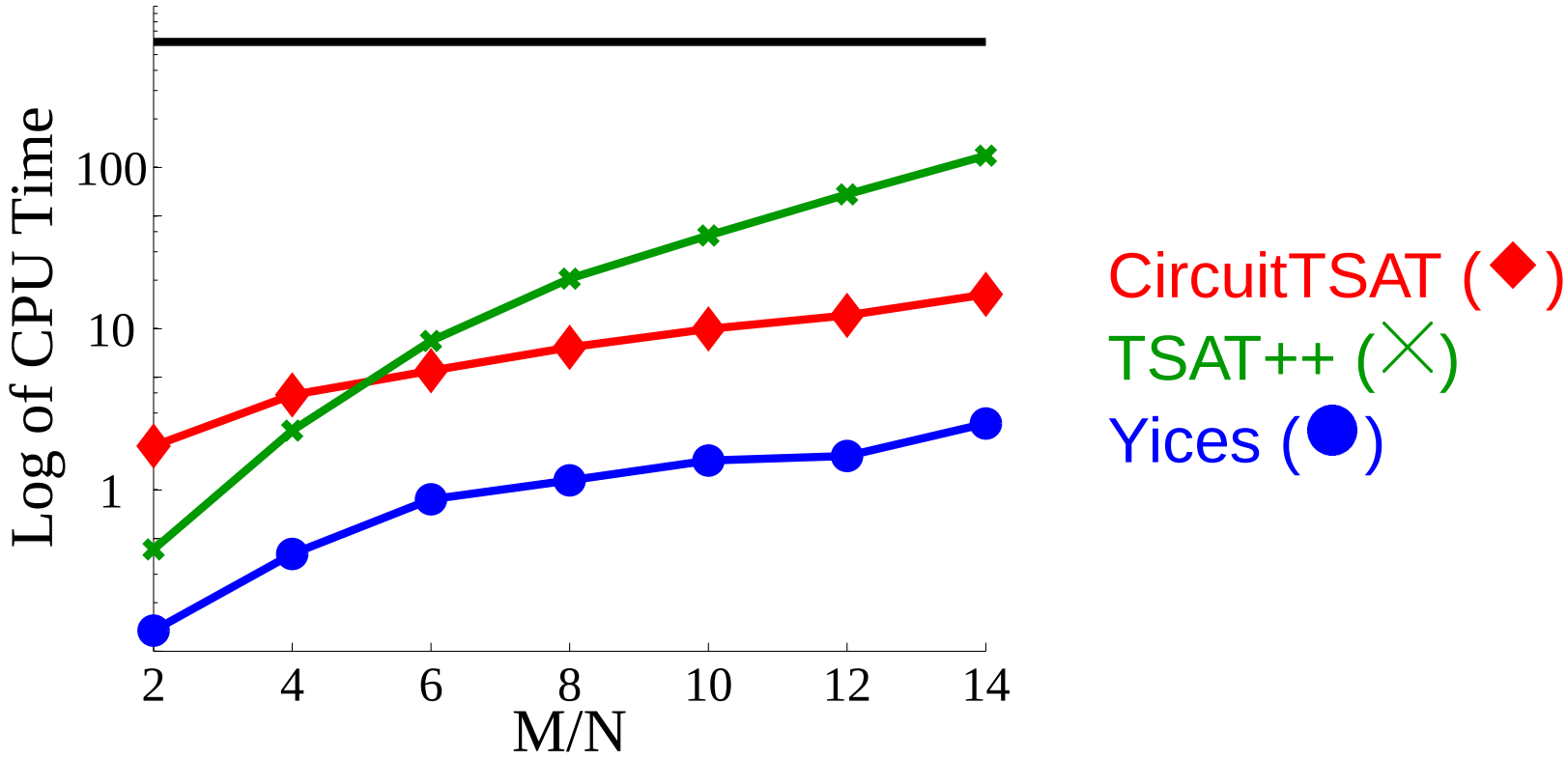
TSAT++ (×)

Yices (●)

Performance Comparison of CircuitTSAT

Increasing Number of Literals

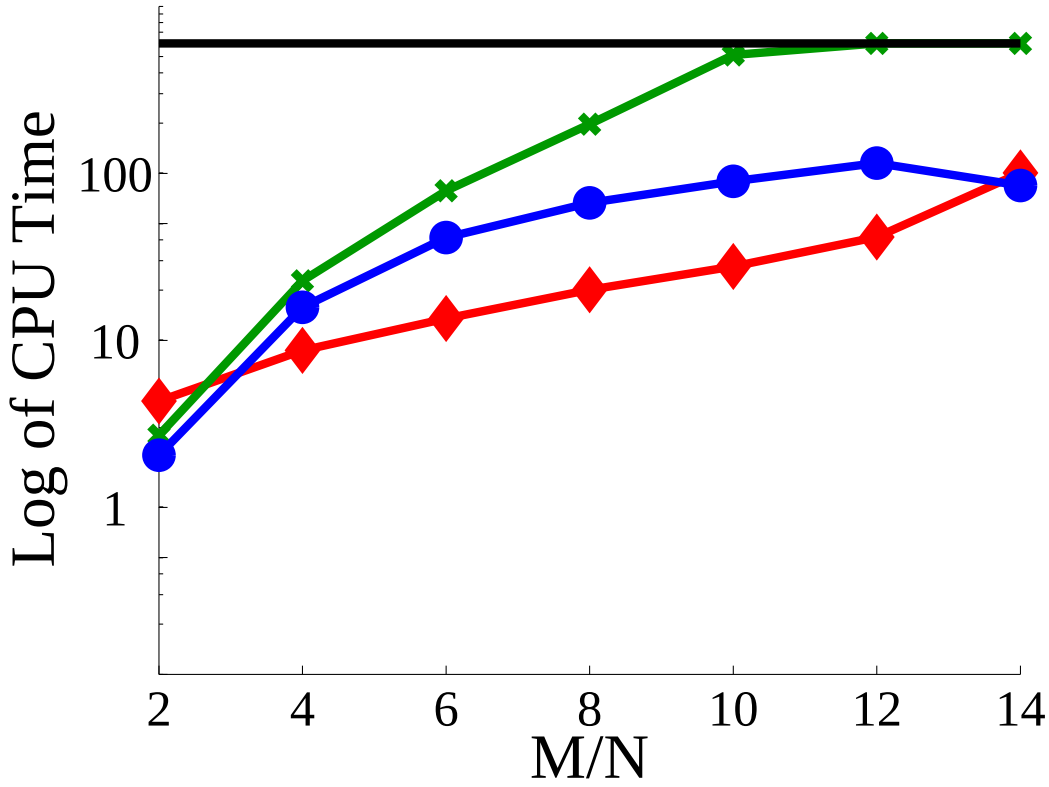
Logarithm of Median Running Times for
$K = 5$ and $N = 50$



Performance Comparison of CircuitTSAT

Increasing Number of Literals

Logarithm of Median Running Times for
$K = 5$ and $N = 100$



CircuitTSAT (◆)

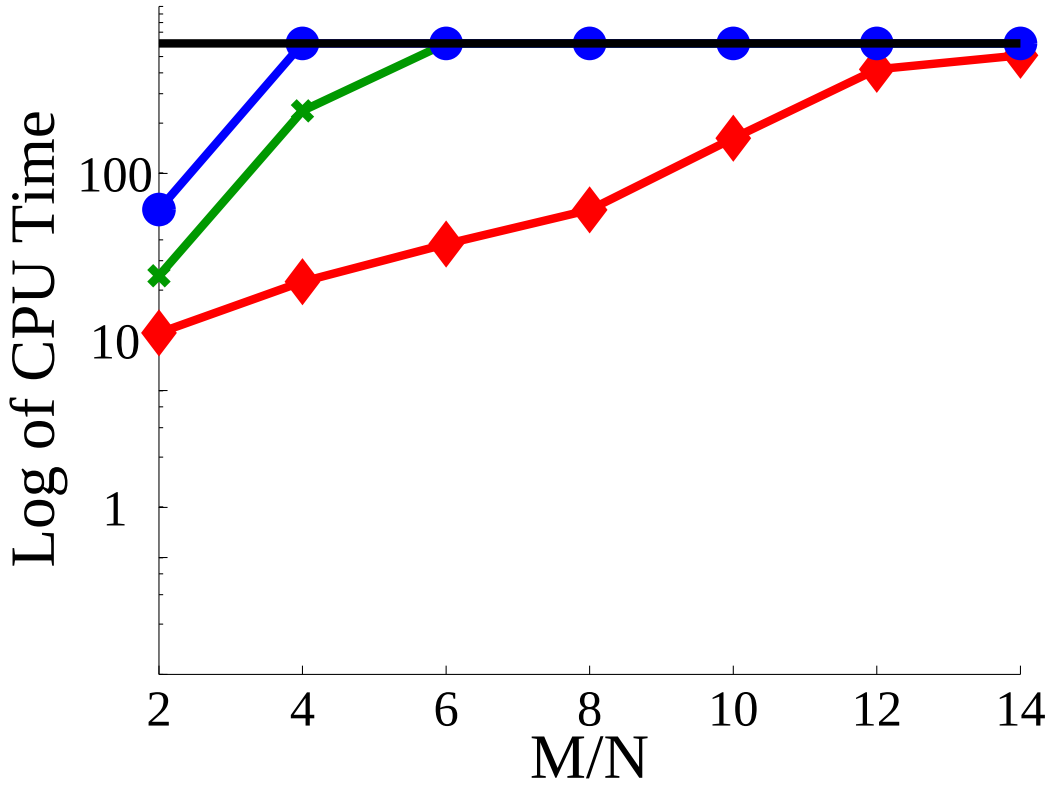
TSAT++ (×)

Yices (●)

Performance Comparison of CircuitTSAT

Increasing Number of Literals

Logarithm of Median Running Times for
$K = 5$ and $N = 200$



CircuitTSAT (◆)

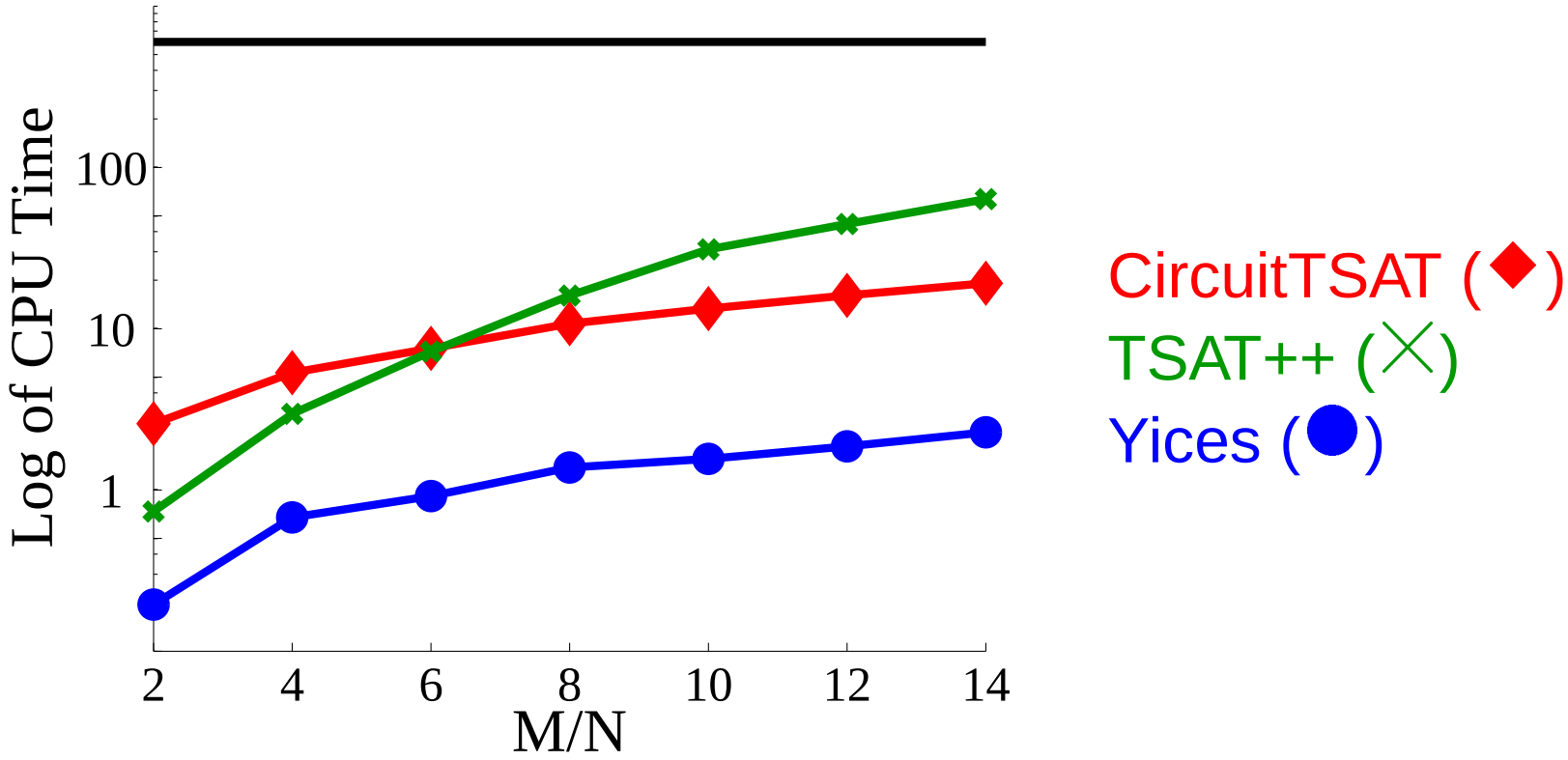
TSAT++ (×)

Yices (●)

Performance Comparison of CircuitTSAT

Increasing Number of Literals

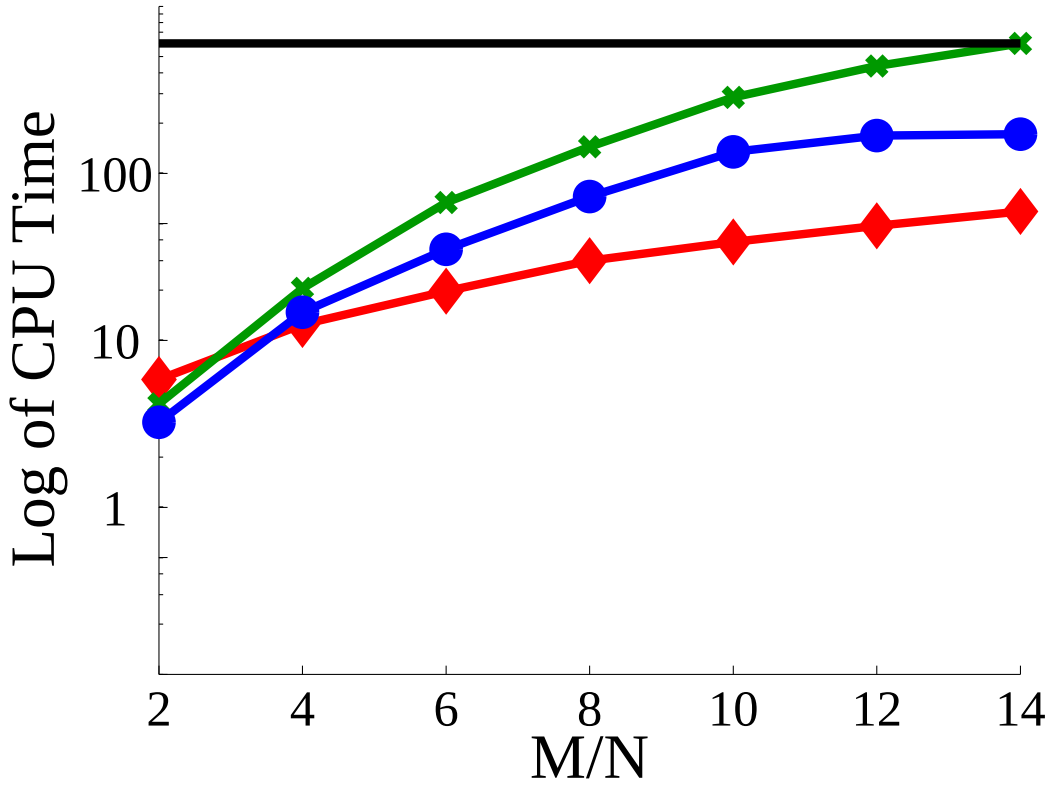
Logarithm of Median Running Times for
$K = 7$ and $N = 50$



Performance Comparison of CircuitTSAT

Increasing Number of Literals

Logarithm of Median Running Times for
$K = 7$ and $N = 100$



CircuitTSAT (◆)

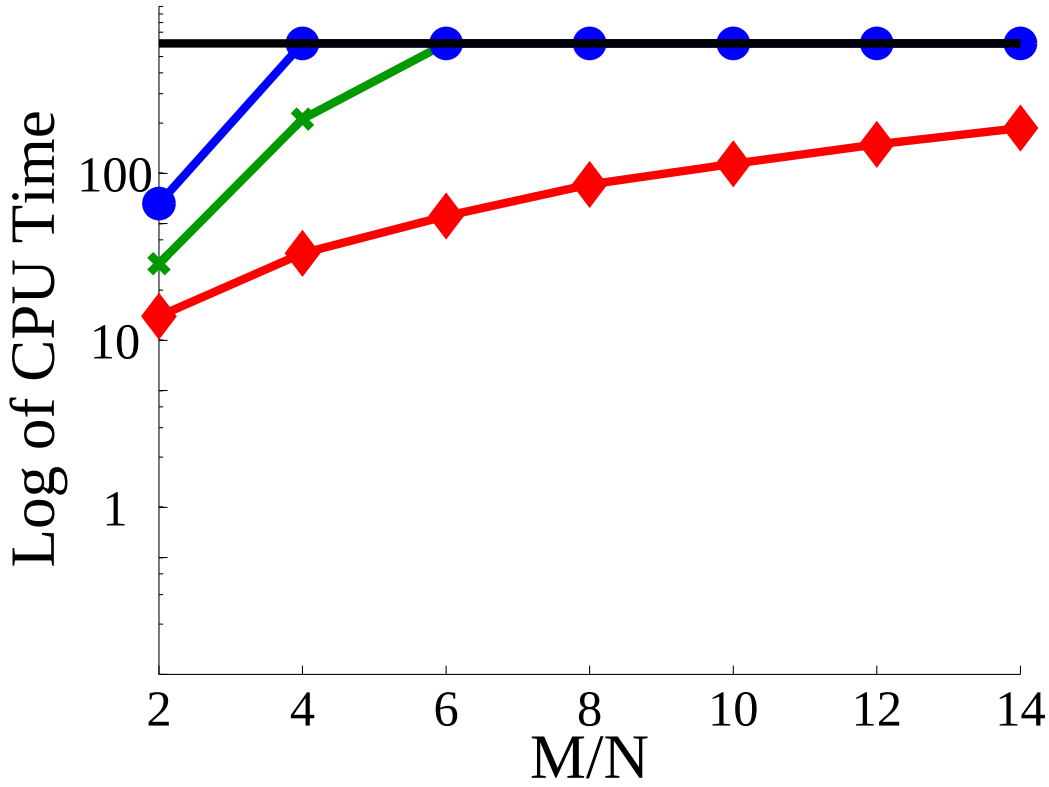
TSAT++ (×)

Yices (●)

Performance Comparison of CircuitTSAT

Increasing Number of Literals

Logarithm of Median Running Times for
$K = 7$ and $N = 200$



CircuitTSAT (◆)

TSAT++ (×)

Yices (●)

Future Work
Extending Circuit-based Approaches to DTPs

- Preprocessing steps similar to TSAT++
- Better strategies for allocating bits
  - Using q_{max} bits is a conservative allocation approach
  - May be able to allocate each X_i with different size
  - May be able to employ over-approximation of [Bryant et al, 2007]
- Expanding to Hybrid CSPs
  - Extending CircuitTSAT to hybrid constraints
  - Applications to more problems in planning and scheduling
- Better SAT solvers
  - SAT solvers could be designed to better exploit the structural information from a DTP
  - Similarly the MODOC solver was designed for planning [Gelder and Okushi, 1999]

Conclusion

- We demonstrated the capabilities of CircuitTSAT — a circuit-based approach for solving DTP instances
- We compared the circuit-based approach to other DTP and SMT solvers
  - Yices faster for smaller values of K and N
  - For large K and N, CircuitTSAT scales better and significantly outperformed both Yices and TSAT++
  - CircuitTSAT exploits structural information of DTPs
- It remains to be seen if circuit-based approaches can be integrated with other approaches for better overall performance

The End

Any question?

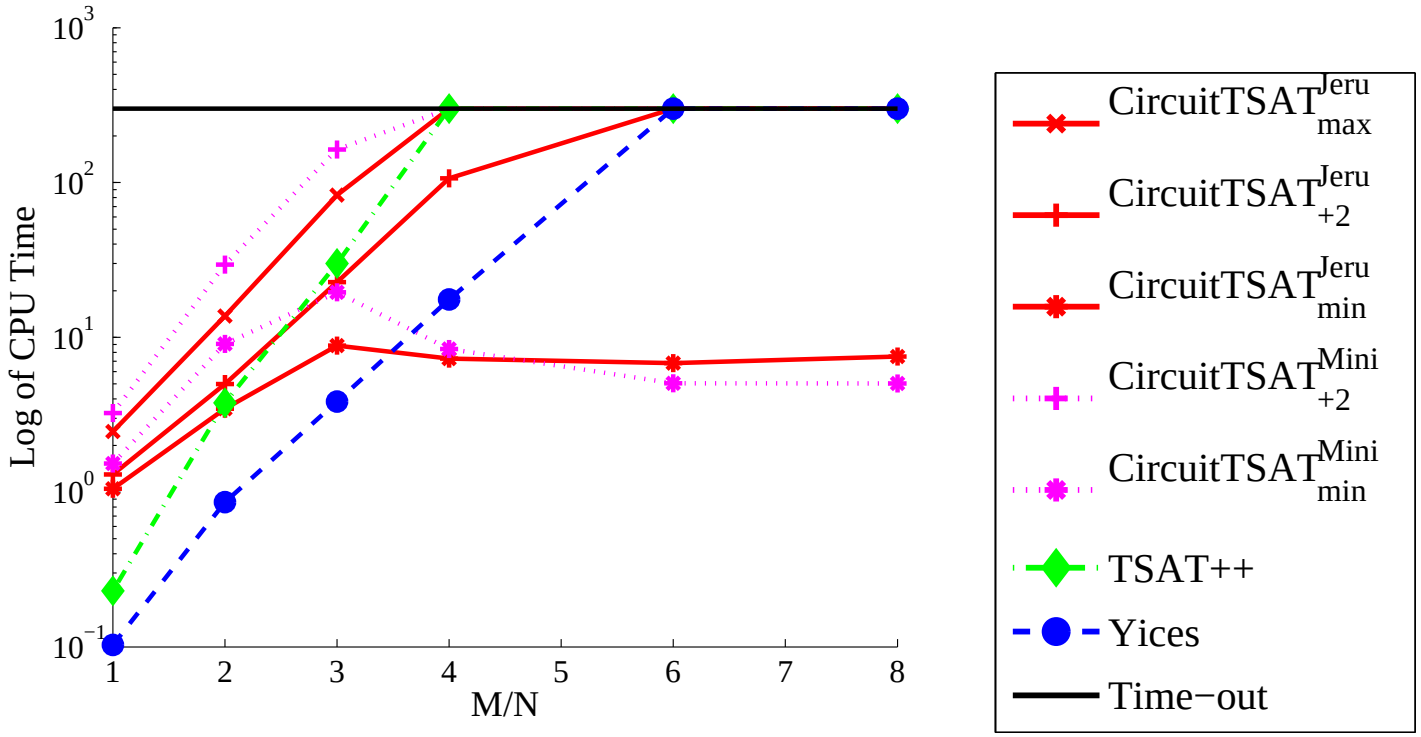
Experiment 2
Comparing Performance for Different Variants of TSAT

- Here we compare the execution times of variants of CircuitTSAT
- We explore performance for large numbers of temporal variables (N) and many disjuncts per clause (K)
- In the following plots, we plot log running time vs. clause to variable ratio (M/N)
- These tests demonstrate the effectiveness of different SAT solvers for our approach and the advantages of using a smaller bit-space size q.

Performance Comparison of CircuitTSAT

Variants of CircuitTSAT

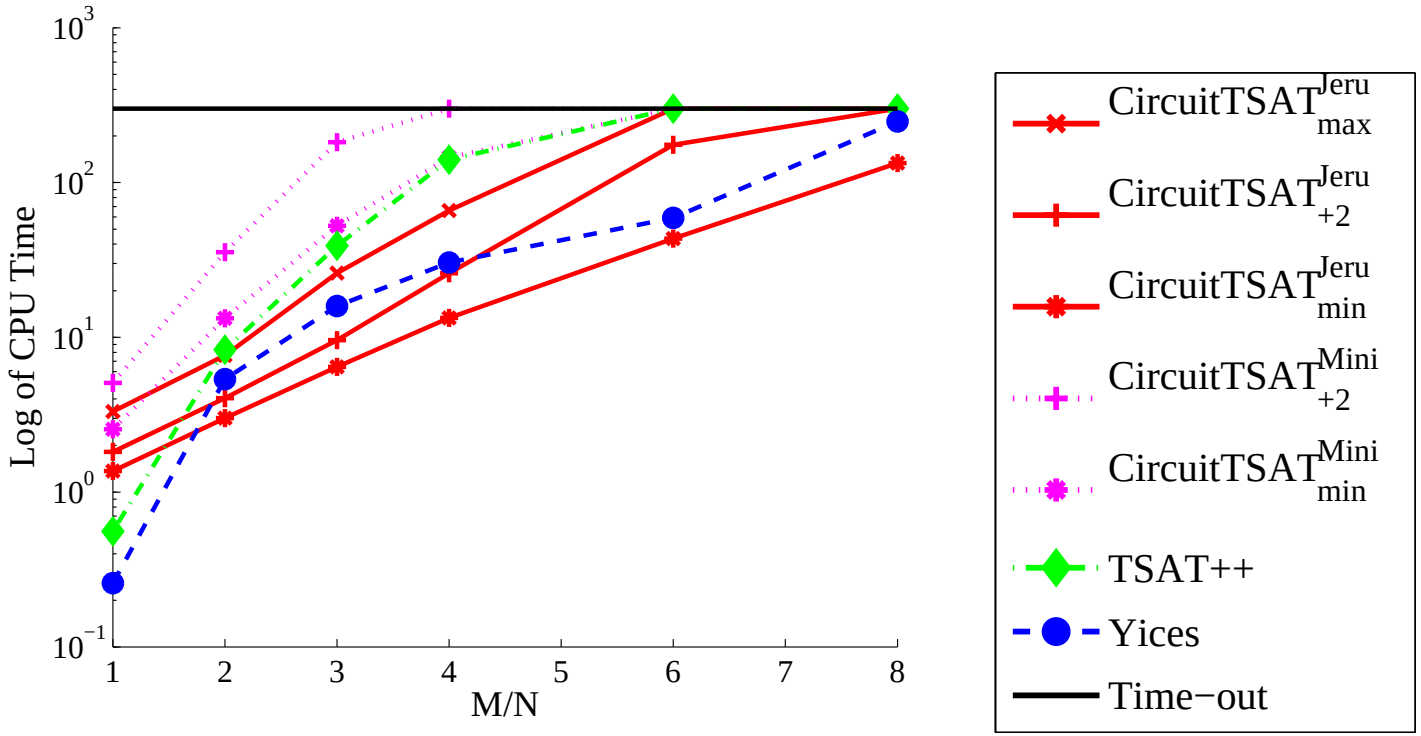
Logarithm of Median Running Times for $N = 150$ and $K = 2$



Performance Comparison of CircuitTSAT

Variants of CircuitTSAT

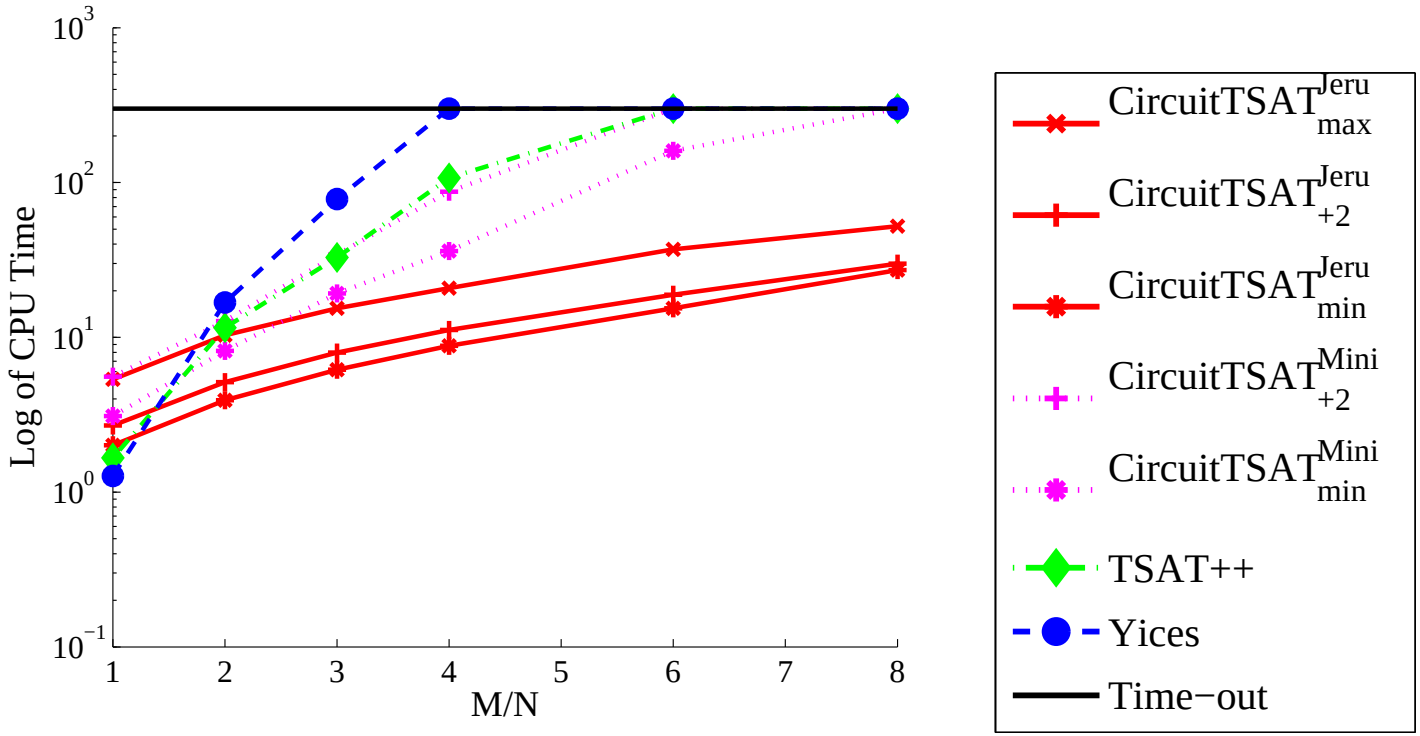
Logarithm of Median Running Times for $N = 150$ and $K = 3$



Performance Comparison of CircuitTSAT

Variants of CircuitTSAT

Logarithm of Median Running Times for $N = 150$ and $K = 5$



Extra Slides
DTP Notation

- DTC $\mathcal{D}$ and a DTP $\mathcal{P}$ then expressed as follows:

$$\mathcal{D}_{\Lambda_\ell} = \bigvee_{(\lambda_i,\gamma_i,z_i)\in\Lambda_\ell} \mathcal{S}_{\gamma_i,\lambda_i}(z_i) \;; \qquad \mathcal{P} = \bigwedge_{\ell=1}^{M} \mathcal{D}_{\Lambda_\ell}$$

where Λ_ℓ is a set containing triplets $(\lambda_i, \gamma_i, z_i)$

Extra Slides
Handling Overflows

- $X - Y + z \geq 0$ is expressed by c_q and d_q but it can overflow
- There are 2 cases:
  1. $z \leq 0$:
     - $d_q = 0$ implies that $X - Y < 0$ hence $\bar{d}_q \Rightarrow X - Y + z < 0$
     - $d_q = 1$ implies that $X - Y \geq 0$ so an overflow can't occur. Hence, $c_q \Leftrightarrow X - Y + z \geq 0$.

     $$(X - Y + z \geq 0) \quad \Leftrightarrow \quad (d_q \wedge c_q)$$

  2. $z > 0$:
     - $d_q = 1$ implies that $X - Y \geq 0$ hence $d_q \Rightarrow X - Y + z \geq 0$.
     - $d_q = 0$ implies that $X - Y < 0$ so an overflow can't occur. Hence, $c_q \Leftrightarrow X - Y + z \geq 0$.

     $$(X - Y + z \geq 0) \Leftrightarrow (d_q \vee c_q)$$

- We can represent the DTCs state by $d_q \wedge c_q$ or

Extra Slides
CNF for d_i carry bits

- The CNF for d_i is

$$
\begin{array}{lll}
(d_i \vee d_{i-1} \vee \bar{x}_i \vee y_i) & \wedge & (d_i \vee \bar{d}_{i-1} \vee y_i) \quad \wedge \\
(d_i \vee \bar{d}_{i-1} \vee \bar{x}_i) & \wedge & (\bar{d}_i \vee \bar{d}_{i-1} \vee x_i \vee \bar{y}_i) \quad \wedge \\
(\bar{d}_i \vee d_{i-1} \vee \bar{y}_i) & \wedge & (\bar{d}_i \vee d_{i-1} \vee x_i)
\end{array}
$$

Extra Slides

CNF for c_i carry bits

- The CNF for c_i is

condition A	condition B
$c_i \vee \bar{c}_{i-1}$	$\bar{c}_i \vee c_{i-1}$
$c_i \vee c_{i-1} \vee d_{i-1} \vee x_i \vee y_i$	$c_i \vee \bar{c}_{i-1} \vee d_{i-1} \vee x_i \vee y_i$
$c_i \vee c_{i-1} \vee d_{i-1} \vee \bar{x}_i \vee \bar{y}_i$	$c_i \vee \bar{c}_{i-1} \vee d_{i-1} \vee \bar{x}_i \vee \bar{y}_i$
$c_i \vee c_{i-1} \vee \bar{d}_{i-1} \vee \bar{x}_i \vee y_i$	$c_i \vee \bar{c}_{i-1} \vee \bar{d}_{i-1} \vee \bar{x}_i \vee y_i$
$c_i \vee c_{i-1} \vee \bar{d}_{i-1} \vee x_i \vee \bar{y}_i$	$c_i \vee \bar{c}_{i-1} \vee \bar{d}_{i-1} \vee x_i \vee \bar{y}_i$
$\bar{c}_i \vee c_{i-1} \vee d_{i-1} \vee \bar{x}_i \vee y_i$	$\bar{c}_i \vee \bar{c}_{i-1} \vee d_{i-1} \vee \bar{x}_i \vee y_i$
$\bar{c}_i \vee c_{i-1} \vee d_{i-1} \vee x_i \vee \bar{y}_i$	$\bar{c}_i \vee \bar{c}_{i-1} \vee d_{i-1} \vee x_i \vee \bar{y}_i$
$\bar{c}_i \vee c_{i-1} \vee \bar{d}_{i-1} \vee x_i \vee y_i$	$\bar{c}_i \vee \bar{c}_{i-1} \vee \bar{d}_{i-1} \vee x_i \vee y_i$
$\bar{c}_i \vee c_{i-1} \vee \bar{d}_{i-1} \vee \bar{x}_i \vee \bar{y}_i$	$\bar{c}_i \vee \bar{c}_{i-1} \vee \bar{d}_{i-1} \vee \bar{x}_i \vee \bar{y}_i$

Extra Slides

References I

- Armando A., Castellini C. and Giunchiglia E. (2000).
  SAT-Based Procedures for Temporal Reasoning.
  In *Procs. of ECP'00*.
- Armando A., Castellini C., Giunchiglia E. and Maratea M. (2004).
  A SAT-Based Decision Procedure for the Boolean Combination of Difference Constraints.
  In *Procs. of SAT'04*.
- Benton J., Do B. and Kambhampati S. (2005).
  Over-Subscription Planning with Numeric Goals.
  In *Procs. of IJCAI'05*.

References II

- Bryant R., Lahiri S. and Seshia S. (2002).
  Modeling and Verifying Systems Using a Logic of Counter Arithmetic with Lambda Expressions and Uninterpreted Functions.
  In *CAV'02: 14th International Conference on Computer Aided Verification*.
- Bryan R., Kroening D., Ouaknine J., Seshia S., Strichman O. and Brady B. (2007).
  Deciding Bit-Vector Arithmetic with Abstraction
  In .
- Dechter R., Meiri I. and Pearl J. (1991).
  Temporal Constraint Networks.
  Artificial Intelligence 49(1–3).

References III


Do B. and Kambhampati S. (2003).
SAPA: A Multi-Objective Metric Temporal Planner.
J. Artif. Intell. Res. (JAIR) 20.


Dutertre B. and de Moura L. (2006).
The Yices SMT solver.
http://yices.csl.sri.com/tool-paper.pdf.


Eén N. and Sörensson N. (2003).
An Extensible SAT-Solver.
In *Proceedings of SAT'03*.

References IV

- Gelder A. and Okushi F. (1999).
  A Propositional Theorem Prover to Solve Planning and Other Problems.
  Annals of Mathematics and AI 26(1-4).
- Heuring V., Jordan H. and Murdocca M. (1997).
  Computer Systems Design and Architecture.
  Addison-Wesley.
- Kumar T. K. S. (2003).
  Incremental Computation of Resource-Envelopes in Producer-Consumer Models.
  In *Procs. of CP'03*.

References V

- Kumar T. K. S. (2005).
  On the Tractability of Restricted Disjunctive Temporal Problems.
  In *Procs. of ICAPS'05*.
- Kumar T. K. S. (2006).
  Tractable Classes of Metric Temporal Problems with Domain Rules.
  In *Procs. of AAAI'06*.
- Moskewicz M., Madigan C., Zhao Y., Zhang L. and Malik S. (2001).
  CHAFF: Engineering an Efficient SAT Solver.
  In *Design Automation Conference'01*.

References VI


Nadel A. (2002).
Backtrack Search Algorithms for Propositional Logic Satisfiability: Review and Innovations.
Master's Thesis, Hebrew University of Jerualem.


Stergiou K. and Koubarakis M. (1998).
Backtracking Algorithms for Disjunctions of Temporal Constraints.
In *Procs. of AAAI'98*.


Stergiou K. and Koubarakis M. (2000).
Backtracking Algorithms for Disjunctions of Temporal Constraints.
Artificial Intelligence.

References VII

Tsamardinos I. and Pollack M. (2003).
Efficient Solution Techniques for Disjunctive Temporal Reasoning Problems.
Artificial Intelligence 151(1-2).