Fast and Flexible Difference Constraint Propagation for DPLL(T)

Scott Cotton Oded Maler

Verimag
Centre Equation
Grenoble, France

SAT, 2006

Outline

Introduction

SMT

- SMT solvers determine the satisfiability of a Boolean combination of *predicates*.
- The predicates fall in some background *theory*, such as linear real arithmetic.
- Very simple theories can be useful.

Lazy SMT

- Works within the DP framework.
- DP interprets predicates as propositional variables.
- Integrates an interpreter $\mathbb{I}$ for a theory for consistency checking of truth assignments and constraint propagation.

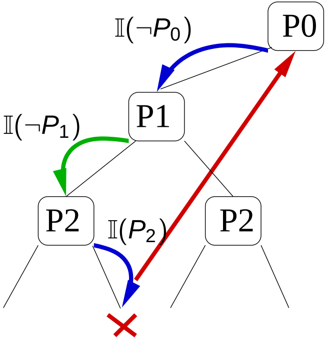
Introduction – Lazy SMT and Theory Propagation

Deduced by $\mathbb{I}$

Deduced by DPLL

```
-P0
 P0 + -P1 + P2
-P1 + -P2
 P0 +  P1 + P2
```

$$P_0 \stackrel{def}{=} x > 0$$
$$P_1 \stackrel{def}{=} x > 1$$
$$P_2 \stackrel{def}{=} x > 2$$



Introduction – Contributions

1. A framework for *flexibility* of constraint propagation, in any theory.
2. Optimization of constraint propagation for *difference logic*.

Motivating different propagation priorities.

- Constraint propagation is *interleaved* with unit propagation.
- Constraint propagation may be more or less expensive than unit propagation.
- Both methods of propagation can deduce the same predicates.
- If a dead end can be found by one propagation method alone, the other need not be called.

Constraint Labels can be used to maintain state with respect to theory propagation.

Constraint Labels for Propagation Roles

- Π A set of assigned constraints whose consequences have been found.
- Σ All assigned constraints whose consequences have not been found.
- Δ A set of assigned or unassigned constraints which are consequences of the constraints labelled Π.
- Λ All other constraints (unassigned).

A Theory Software Interface.

- SetTrue: Add a predicate p to the current truth assignment.
  - If $p \in \Delta$, ignore it.
  - If $p \in \Lambda$, label it Σ and check whether $\Pi \cup \Sigma$ is T-consistent.
- TheoryProp: Find and justify some consequences of the current truth assignment:
  - Pick a constraint $p \in \Sigma$, label it Π.
  - Find (and justify) consequences c of Π such that $c \notin \Delta$.
  - For every consequence c, if $c \in \Lambda$, inform DP c is a new consequence. Label every c as Δ.
- Backtrack: Remove some predicates from the current truth assignment:
  - Label all newly unassigned constraint Λ.
  - Label any unassigned constraints in Δ as Λ.

Implementing Interleaving Strategies

- The labels allow propagation to compute consequences of all assigned constraints by finding consequences of only Π-labelled constraints.
- Theory interface decouples propagation from DP assignments, allowing TheoryProp to be called at various times in DP procedure.

Two interleaving strategies

- Lazy propagation. Only call TheoryProp when DP has no unit implications.
- Eager propagation. Call TheoryProp with every call to SetTrue.

Optimizing Difference Constraint Propagation

About Difference Constraints

- Difference constraints are constraints in the form $x - y \leq c$.
- They are applicable to many scheduling and timing analysis problems.
- Conjunctions of difference constraints have a convenient graphical representation.

Definition (Constraint graph)

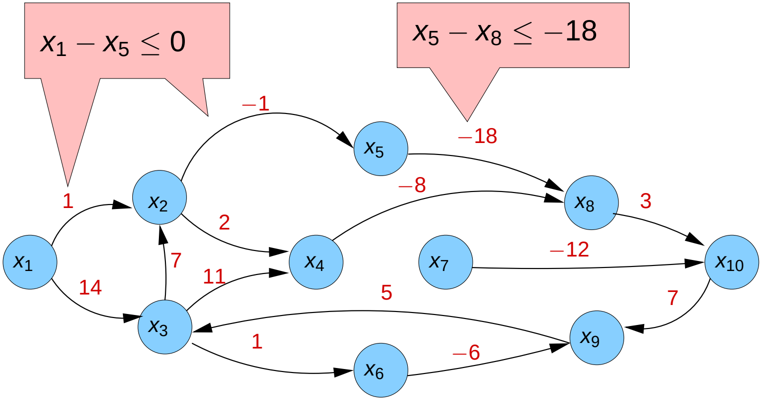
Let S be a set of difference constraints and let G be the graph comprised of one weighted edge $x \xrightarrow{c} y$ for every constraint $x - y \leq c$ in S. We call G the constraint graph of S.

Theorem

Let Γ be a conjunction of difference constraints, and let G be the constraint graph of Γ. Then Γ is satisfiable if and only if there is no negative cycle in G. Moreover, if Γ is satisfiable, then $\Gamma \models x - y \leq c$ if and only if y is reachable from x in G and $c \geq d_{xy}$ where d_{xy} is the length of a shortest path from x to y in G.

Constraint Graphs
Example

Example Constraint Graph



Definition (Potential Function)

Given a weighted directed graph $G = (V, E, W)$, a *potential function* π is a function $\pi : V \rightarrow \mathbb{R}$ such that $\pi(x) + W(x, y) - \pi(y) \geq 0$ for every edge $(x, y) \in E$.

Some Potential function properties

- A potential function exists iff G contains no negative cycle.
- Given a potential function π for a constraint graph G, a satisfying assignment σ for the set of difference constraints in G is given by $\sigma(x) \mapsto -\pi(x)$.

Incremental Consistency Checking

An algorithm

SetTrue($u - v \leq d$):

Let $G = \Pi \cup \Sigma$. Given a potential function π for G, find a potential function π' for the graph $G \cup \{u \xrightarrow{d} v\}$ if one exists.

An $\mathcal{O}(m + n \log n)$ algorithm:

$$
\boxed{
\begin{array}{l}
\gamma(v) \leftarrow \pi(u) + d - \pi(v) \\
\gamma(w) \leftarrow 0 \text{ for all } w \neq v \\
\textbf{while } \min(\gamma) < 0 \wedge \gamma(u) = 0 \\
\quad s \leftarrow \operatorname{argmin}(\gamma) \\
\quad \pi'(s) \leftarrow \pi(s) + \gamma(s) \\
\quad \gamma(s) \leftarrow 0 \\
\quad \textbf{for } s \xrightarrow{c} t \in G \textbf{ do} \\
\quad\quad \textbf{if } \pi'(t) = \pi(t) \textbf{ then} \\
\quad\quad\quad \gamma(t) \leftarrow \min\{\gamma(t), \pi'(s) + c - \pi(t)\}
\end{array}
}
$$

TheoryProp Outer loop

Repeat until no constraints are labelled Σ or until DP is notified of a new consequence:

1. Pick a constraint c labelled Σ and find the consequences S of $\Pi \cup \{c\}$ which are not consequences of Π.
2. Notify DP of any consequences in S which are labelled Λ.
3. Relabel c with Π and every constraint in S with Δ.

TheoryProp Outer loop

Repeat until no constraints are labelled Σ or until DP is notified of a new consequence:

1. Pick a constraint c labelled Σ and find the consequences S of $\Pi \cup \{c\}$ which are not consequences of Π.
2. Notify DP of any consequences in S which are labelled Λ.
3. Relabel c with Π and every constraint in S with Δ.

TheoryProp Inner loop

Find consequences of $\Pi \cup \{(x - y \leq c)\}$ which are not consequences of Π.

1. Compute single source shortest paths (SSSP) $\delta^{\rightarrow}$ in constraint graph of Π starting from y.
2. Compute SSSP $\delta^{\leftarrow}$ in *reversed* constraint graph Π starting from x.
3. For every constraint $u - v \leq d$ labelled Λ or Σ, if $\delta^{\leftarrow}(u) + c + \delta^{\rightarrow}(v) \leq c$ then $u - v \leq d$ is a consequence.

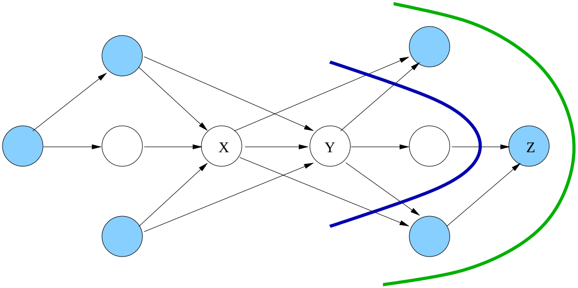
(due to Nieuwenhaus et al CAV'04)

An Observation

1. The best SSSP computations on arbitrarily weighted graphs are $\mathcal{O}(mn)$.
2. The potential function computed during consistency checking is a potential function for the constraint graph of Π.
3. A potential function can be used to translate a shortest path problem for *arbitrarily* weighted graphs into a shortest path problem on *non-negatively* weighted graphs.
4. The best SSSP computations on non-negatively weighted graphs are atleast as good as $\mathcal{O}(m + n \log n)$.

Do we need the entire SSSP results $\delta^{\rightarrow}$ and $\delta^{\leftarrow}$?

When finding consequences of $\Pi \cup \{x - y \leq c\}$, if the shortest path from y to some vertex z is atleast as short as the shortest path from x to z, then any constraint $u - z \leq d$ is not a new consequence:

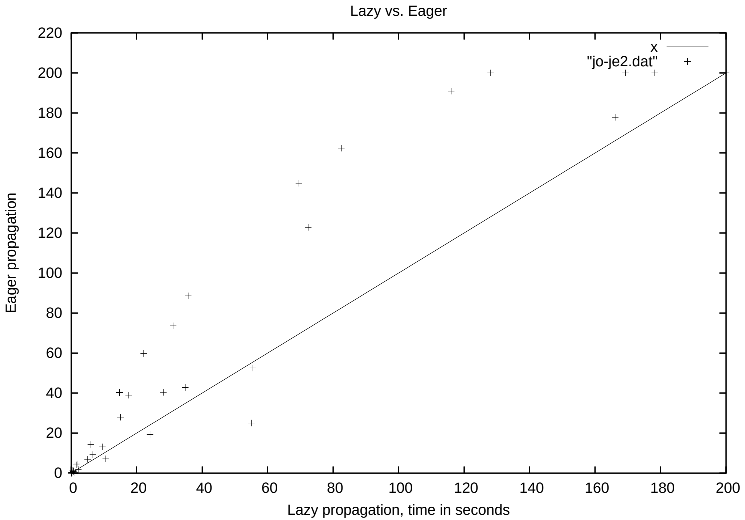


Experimental Results

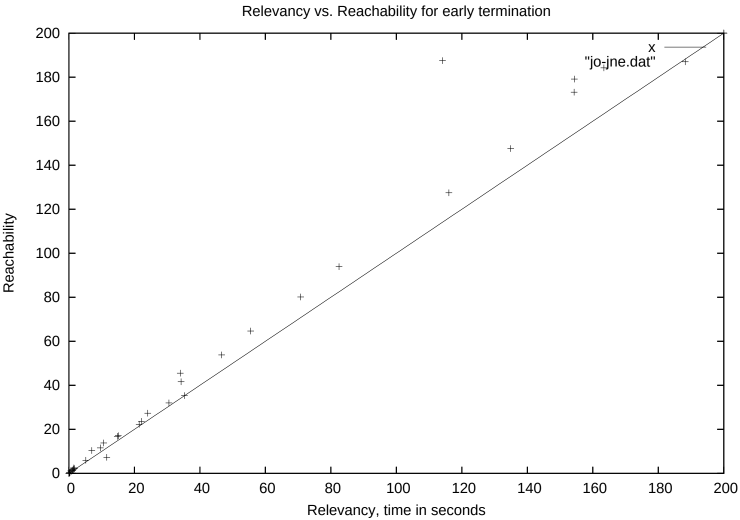
Experiments

- All experiments performed on job shop scheduling problems.
- These problems are strongly constrained by difference constraints and weakly propositionally constrained.
- These problems stress test difference constraint propagation in a lazy SMT framework.

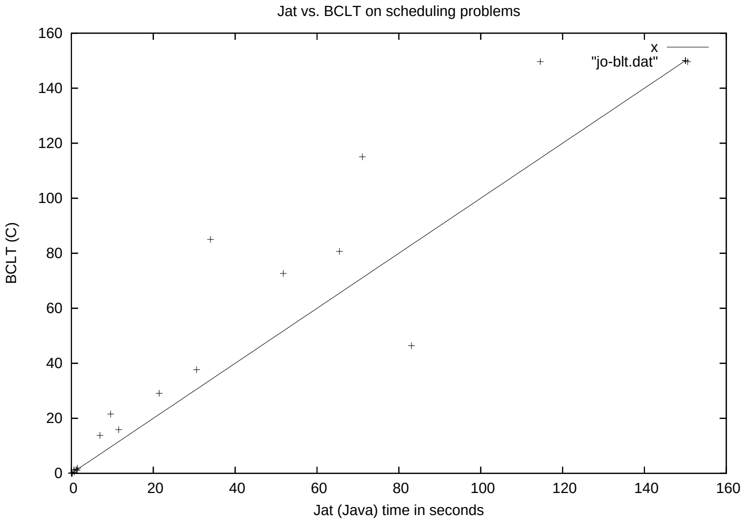
Eager *v* Lazy Propagation



Reachable *v* Relevancy



Jat *v* Barcelogic Tools



Conclusion

- Lazy propagation is easy to implement with constraint labels, and experiments show it is a good propagation strategy.
- Complete difference constraint propagation can be achieved in $\mathcal{O}(m + n \log n + |U|)$ time.
- Relevancy based early termination is helpful.

Thankyou! (and Questions?)