Constraint Propagation in Propositional Planning

Andreas Sideris and **Yannis Dimopoulos**
Department of Computer Science
University of Cyprus
1678 Nicosia, Cyprus
sideris@cytanet.com.cy, yannis@cs.ucy.ac.cy

Abstract

Planning as Satisfiability is a most successful approach to optimal propositional planning. It draws its strength from the efficiency of state-of-the-art propositional satisfiability solvers, combined with the utilization of constraints that are inferred from the problem planning graph. One of the recent improvements of the framework is the addition of long-distance mutual exclusion (londex) constraints that relate facts and actions which refer to different time steps.

In this paper we compare different encodings of planning as satisfiability wrt the constraint propagation they achieve in a modern SAT solver. This analysis explains some of the differences observed in the performance of different encodings, and leads to some interesting conclusions. For instance, the BLACKBOX encoding achieves more propagation than the one of SATPLAN06, and therefore is a stronger formulation of planning as satisfiability. Moreover, our investigation suggests a new more compact and stronger model for the problem. We prove that in this new formulation many of the londex constraints are redundant in the sense that they do not add anything to the constraint propagation achieved by the model. Experimental results suggest that the theoretical results obtained are practically relevant.

Introduction

One of the most successful approaches to optimal STRIPS planning is the SATPLAN approach, that translates a planning problem to a propositional satisfiability one (SAT). The approach draws its strength from the efficiency of state-of-the-art propositional satisfiability solvers, combined with the utilization of constraints that are inferred from the problem planning graph. After more than a decade of research, there exist nowadays many different encodings of propositional planning as satisfiability, including those of BLACKBOX (Kautz and Selman 1999) and SATPLAN06 (Kautz, Selman, and Hoffmann 2006). In most of the studies these formulations are compared experimentally, and little is known about their theoretical underpinnings and the reasons that render one model better than the other. In this work we present a first *theoretical analysis* that compares some of these encodings and explains important reasons that contribute to the differences which are observed in their performance.

Our investigation is based on the simple observation that the planning as satisfiability framework regards the planning problem as a Constraint Satisfaction one. Therefore, *constraint propagation*, i.e. the process of deriving the values of new variables from values that are known or assumed for other variables, is a central notion. Stronger forms of propagation derive more variable values and therefore lead to more pruning of the search space than weaker ones. If the computational cost of the constraint propagation procedure is low, the reduction of the search space usually translates into better run times.

This work compares different planning encodings wrt the *unit propagation* they achieve, the standard constraint propagation method employed in almost all modern SAT solvers. Roughly speaking, a planning model is stronger than another if it is able to propagate more variable values. Moreover, one encoding is more compact than some other if it achieves the same propagation but with a subset of the clauses. The clauses that are contained in the less compact encoding are *redundant* wrt unit propagation.

Our analysis reveals some interesting relationships. The most unexpected is probably that the BLACKBOX encoding is stronger than the one used in SATPLAN06. Based on our theoretical results we propose a new encoding of planning as satisfiability, called SAT-MAX-PLAN (abbreviated as SMP), that achieves more propagation than all other models, and it does so with a set of clauses that contains no redundancy.

We also study the propagation power of *long distance mutual exclusion constraints* (londex), as introduced in the MAXPLAN system (Chen, Xing, and Zhang 2007), and show that they indeed strengthen the model of the SATPLAN06 encoding. More precisely, we prove that SATPLAN06 can propagate londex type information forward through the layers of the propositional theory, i.e. from variables that refer to a time point to variables that refer to some later time point. However, SATPLAN06 fails to do the same backwards, and therefore adding londex type constraints to SATPLAN06 encodings improves propagation. However, we show that londex constraints do not increase the propagation of the SMP encoding, and are therefore redundant in this new model.

In the experimental part we compare SMP, BLACKBOX and SATPLAN06 in a number of domains from the last planning competitions. It turns out that SMP outperforms both other encodings, whereas between the two, BLACKBOX has an advantage over SATPLAN06. In fact, SMP coupled with a new SAT solver called precosat (Biere 2009), can solve more problems than the other planners, and presents a notable advancement of the state-of-the-art of planning as satisfiability. Moreover, it shows that the theoretical results of this work are of practical relevance.

To the best of our knowledge the only other studies close to the spirit of this work are (Geffner 2004) and (Rintanen 2008). However, the focus there is on understanding mutexes and londexes, and explaining how they can be derived by a modern SAT solver. Our investigation is complementary to the above, and explains *what*, *when* and *why* constraints improve performance.

Preliminaries

In this section we discuss briefly various SAT encodings for STRIPS planning problems, constraint propagation in SAT, and long distance mutual exclusion constraints. We assume that the reader is familiar with the basics of STRIPS planning, planning graphs, londex constraints, and SAT solving. Our analysis refers exclusively to STRIPS planning.

Planning as Satisfiability

A (STRIPS) planning problem is a triple $P =< I, G, A >$, where I is the set of facts that hold in the initial state, G are the goals, and A is a set of actions. Each action A has preconditions, denoted by $pre(A)$, add effects, denoted by $add(A)$, and delete effects, denoted by $del(A)$. From the description of a planning problem a *planning graph* can be constructed as described in (Blum and Furst 1997). A central notion in planning graphs is action *interference*, defined as follows.

Definition 1 *Two action A_1 and A_2* interfere *whenever any of the sets $del(A_1) \cap add(A_2)$ and $del(A_1) \cap pre(A_2)$ is non-empty.*

In this work we study some of the most successful models of planning as satisfiability. They include different encodings supported by the planning systems BLACKBOX, and SATPLAN06. All these systems use information that is derived from the planning graph of the problem. Part of the information that is extracted has the form of mutually exclusive pairs, or mutexes for short, that are defined as follows.

1. Two actions A_1, A_2 are mutually exclusive at level l if they interfere or there is a pair of facts $f_1 \in pre(A_1)$, $f_2 \in pre(A_2)$ such that f_1, f_2 are mutually exclusive at level l.
2. Two facts f_1, f_2 are mutually exclusive at level l if for every pair of actions A_1, A_2 such that $f_1 \in add(A_1)$, $f_2 \in add(A_2)$, A_1, A_2 are mutually exclusive at level $l - 1$.

In a planning graph each level corresponds to a different time point, while inertia is captured by *noop* actions that encode persistence. In the SAT model of a planning problem time-stamped propositional atoms (or variables) represent the action and facts of the problem. An atom $A(T)$, where A is an action, corresponds to the decision of whether action A is taken or not at time T, and analogously for variables of the form $f(T)$ where f is a fact. In all systems that utilize the planning graph an action/fact variable is introduced to the theory only if the corresponding action/fact node is present in the planning graph. More specifically, the variables are created as follows.

1. Unit clauses for the initial and final state.
2. An action variable $A(T)$ is added to the theory if for each $p \in pre(A)$, $p(T)$ occurs in the theory, and there is no pair $p_1, p_2 \in pre(A)$ s.t. $\neg p_1(T) \vee \neg p_2(T)$ belongs to the theory.
3. A proposition variable $p(T)$ is added to the theory if some action variable $A(T-1)$ is in the theory, with $p \in add(A)$.

To facilitate our study we first introduce a new encoding called *Graphplan-direct*, that is direct translation of the planning graph structure into propositional logic. The other encodings that we investigate in the rest of this paper are subsets of the clause set of the Graphplan-direct formulation. The clauses of the Graphplan-direct encoding are the following.

1. Unit clauses for the initial and final state.
2. $A(T) \rightarrow f(T)$, for every action A and fact f s.t. $f \in pre(A)$.
3. $A(T) \rightarrow f(T+1)$, for every action A and fact f s.t. $f \in add(A)$.
4. $A(T) \rightarrow \neg f(T+1)$, for every action A and fact f s.t. $f \in del(A)$.
5. $f(T) \rightarrow A_1(T-1) \vee \ldots \vee A_m(T-1)$, for every fact f and all actions A_i, $1 \leq i \leq m$ (including the noops) s.t. $f \in add(A_i)$.
6. $\neg f(T) \rightarrow A_1(T-1) \vee \ldots \vee A_m(T-1) \vee \neg f(T-1)$, for every fact f and all actions A_i, $1 \leq i \leq m$ s.t. $f \in del(A_i)$.

7.1 $\neg A_1(T) \vee \neg A_2(T)$, for every pair of actions A_1, A_2 such that the set $del(A_1) \cap pre(A_2)$ is non-empty.

7.2 $\neg A_1(T) \vee \neg A_2(T)$, for every pair of actions A_1, A_2 such that the set $del(A_1) \cap add(A_2)$ is non-empty.

7.3 $\neg A_1(T) \vee \neg A_2(T)$, if there is a pair of facts $f_1 \in pre(A_1)$, $f_2 \in pre(A_2)$ such that f_1, f_2 are mutually exclusive at time T.

8 $\neg f_1(T) \vee \neg f_2(T)$, for every pair of facts f_1, f_2 that are mutex at time T.

The first system that employed information derived from the planning graph in the propositional encoding of a planning problem was BLACKBOX (Kautz and Selman 1999). BLACKBOX (version 43) supports different encodings, three of which we investigate here and denote by BB-7, BB-31, and BB-32. Each of them is obtained by selecting the appropriate value (7, 31, or 32) of parameter *axioms*. The set of clauses of each of these encodings (which is a subset of the clauses of the Graphplan-direct model) is the following.

1. **BB-7**: Clauses 1, 2, 5, 7.1, 7.2, 7.3
2. **BB-31**: Clauses 1, 2, 3, 4, 5, 7.1, 8
3. **BB-32**: Clauses 1, 2, 3, 4, 5, 7.1, 7.2, 7.3, 8

Similarly to BLACKBOX, SATPLAN06 also supports different encodings. Two of them are mixed action/fact models, and the other two are action-based encodings. Due to space limitations, in this work we restrict ourselves to the mixed models. They are denoted by SATPLAN06-4 and SATPLAN06-3, and are obtained by setting the *encoding* parameter to value 4 and 3 respectively. Each of them contains the following clauses (again numbers refer to the Graphplan-direct model).

1. **SATPLAN06-4**: Clauses 1, 2, 5, 7.1, 7.2, 8
2. **SATPLAN06-3**: Clauses 1, 2, 5, 7.1, 7.2, 7.3, 8

The propositional theory that results from the above encodings, for a fixed number of time steps T_{max}, is given as input to a SAT solver. Any time step $T \leq T_{max}$, is a *valid time point*.

Propagation in a SAT solver

State-of-the-art SAT solvers such as siege (Ryan 2003), minisat (Een and Sorensson 2003) and precosat (Biere 2009), employ Unit Resolution for constraint propagation. That is, they resolve all unit clauses with all other clauses in the theory, and iterate this process until no further unit clause can be derived (or unsatisfiability is proven). In the following, $UP(T)$ denotes the closure of theory T under Unit Propagation. The notion of *UP-redundancy* plays a central role in our analysis, and is defined as follows.

Definition 2 *The binary clause $l_1 \vee l_2$ is* UP-redundant *wrt a theory T iff either $l_1 \vee l_2 \in T$ or $l_j \in UP(T \cup \{\neg l_i\})$, for $i \neq j$ and $i, j \in \{1, 2\}$.*

Below, several notions regarding the relative strength of propositional theories wrt binary clauses are defined.

Definition 3 *Theory T_1 is at* least as strong *as theory T_2 wrt UP and binary clauses, denoted by $T_1 \geq_{UP} T_2$, iff every clause of $T_2 \backslash T_1$ is binary and UP-redundant wrt T_1.*
Theory T_1 is strictly stronger *than theory T_2 wrt UP and binary clauses, denoted by $T_1 >_{SUP} T_2$, iff $T_1 \geq_{UP} T_2$ or $T_1 \supset T_2$ and $T_2 \ngeq_{UP} T_1$.*
Theory T_1 is more compact *than theory T_2 wrt UP and binary clauses, denoted by $T_1 >_C T_2$, iff $T_1 \geq_{UP} T_2$ and $T_1 \subset T_2$.*

Long-Distance Mutual Exclusion

The Long-Distance Mutual Exclusion (londex) method of (Chen, Xing, and Zhang 2007), is based on the *multi-valued domain formulation* (MDF), in which a planning domain is defined over a set $X = (x_1, \ldots, x_n)$ of multi-valued variables, where each x_i has an associated finite domain $\mathcal{D}_i$. If x is a multi-valued variable from X and v a value from its domain, $x = v$ denotes the assignment of v to x. To associate such an assignment $x = v$ with a boolean fact f, we use the notation $f = MDF(x, v)$.

For every multi-valued variable in a planning problem, the method of (Chen, Xing, and Zhang 2007) builds the domain transition graph, from which the fact distances are calculated.

Definition 4 *Given an MDF variable x with domain $\mathcal{D}_x$, its* domain transition graph *(DTG) G_x is a digraph with vertex set $\mathcal{D}_x$ and arc set A_x, such that $(v, v') \in A_x$ iff there is an action o such that $del(o) = v$ and $add(o) = v'$.*

Definition 5 *Given a DTG G_x, the distance from a fact $f_1 = MDF(x, v_1)$ to another fact $f_2 = MDF(x, v_2)$, denoted by $\Delta_{G_x}(f_1, f_2)$, is defined as the minimum distance from vertex v_1 to vertex v_2 in G_x.*

Based on fact distances, londex constraints for facts and actions are derived. In the following, $t(f)$ denotes the time step at which fact f is true, and $t(a)$ the time step at which an action is chosen. Moreover, we say that an action a is associated with a fact f if f appears in $pre(a)$, $add(a)$ or $del(a)$.

Definition 6 *(Fact Londex) Given two boolean facts f_1 and f_2, that correspond to two nodes in a DTG G_x, such that $\Delta_{G_x}(f_1, f_2) = r$, then there is no valid plan in which $0 \leq t(f_2) - t(f_1) < r$.*

There are two classes of actions londex constraints that are defined below.

Definition 7 *(Class A Action Londex) If actions a and b are associated with a fact f, they are mutually exclusive if one of the following holds:*

1. *$f \in add(a), f \in del(b)$, and $t(a) = t(b)$*
2. *$f \in del(a), f \in pre(b)$, and $0 \leq t(b) - t(a) \leq 1$*

Definition 8 *(Class B Action Londex) If action a is associated with fact f_1 and action b with fact f_2, and it is invalid to have $0 \leq t(f_2) - t(f_1) < r$ according to fact londex constraints, then a and b are mutually exclusive if one of the following holds:*

1. *$f \in add(a), f \in add(b)$, and $0 \leq t(b) - t(a) \leq r - 1$*
2. *$f \in add(a), f \in pre(b)$, and $0 \leq t(b) - t(a) \leq r$*
3. *$f \in pre(a), f \in add(b)$, and $0 \leq t(b) - t(a) \leq r - 2$*
4. *$f \in pre(a), f \in pre(b)$, and $0 \leq t(b) - t(a) \leq r - 1$*

The relative strength of the encodings

We first study unit propagation in planning systems that employ information derived from the planning graph. More precisely, we will investigate the relative constraint propagation power of BLACKBOX and the mixed SATPLAN06 encodings.

It is easy to see that the encodings are related as follows: for any (STRIPS) planning problem P, SATPLAN06-4(P) $\subset$ SATPLAN06-3(P) $\subset$ BB-32(P), and BB-7(P) $\subset$ SATPLAN06-3(P). The following proposition shows that some of the mutex clauses are UP-redundant in some encodings.

Proposition 1 *The set of clauses 7.3 is UP-redundant wrt any propositional encoding that contains the set of clauses 2 and 8.*

Proof *Let $\neg A_1(T) \vee \neg A_2(T)$ be a clause with A_1 and A_2 two actions such that there is a pair of facts $f_1 \in pre(A_1)$, $f_2 \in pre(A_2)$ such that f_1, f_2 are mutually exclusive at level T. We will show that $\neg A_2(T) \in UP(T_P \cup \{A_1(T)\})$. From $A_1(T)$ and clause $A_1(T) \rightarrow f_1(T)$ we obtain $f_1(T)$. Since the theory contains axioms 8, it must contain the clause $\neg f_1(T) \vee \neg f_2(T)$. From this clause and $f_1(T)$ we obtain $\neg f_2(T)$, form which, together with $A_2(T) \rightarrow f_2(T)$ we conclude $\neg A_2(T)$. The proof of $\neg A_1(T) \in UP(T_P \cup \{A_2(T)\})$ is symmetric.* ■

A direct consequence of the above proposition is the following relation between the two mixed `SATPLAN06` encodings.

Corollary 1 *For any planning problem P, SATPLAN06-4$(P) >_C$ SATPLAN06-3(P).*

Another result that is stated formally below can be used to simplify theories that contain clause sets 3 and 4.

Proposition 2 *The set of clauses 7.2 is UP-redundant wrt any propositional encoding that contains the set of clauses 3 and 4.*

Proof *Let $\neg A_1(T) \vee \neg A_2(T)$ be a clause with A_1 and A_2 two actions such that at least one of the sets $del(A_1) \cap add(A_2)$ and $del(A_2) \cap add(A_1)$ is non-empty. Assume that $del(A_1) \cap add(A_2) \neq \emptyset$ (the other case is symmetric) and let $f \in del(A_1) \cap add(A_2)$. We will show that $\neg A_2(T) \in UP(T_P \cup \{A_1(T)\})$. The theory contains a clause of the form $A_1(T) \rightarrow \neg f(T+1)$ from which $\neg f(T+1)$ is derived. From this and clause $A_2(T) \rightarrow f(T+1)$, $\neg A_2(T)$ is concluded.* ■

A direct consequence of the above proposition is that encoding BB-32 can be simplified by removing clauses 7.2. Similarly, by proposition 1, clauses 7.3 can also be omitted. Therefore, BB-31$(P) >_C$ BB-32(P), for all problems P.

A similar observation holds for the Graphplan-direct encoding. By removing the UP-redundant clauses 7.2 and 7.3 we obtain SATPLAN^{max} encoding which contains the following clauses:

- **SATPLAN^{max}**: Clauses 1, 2, 3, 4, 5, 6, 7.1, 8.

On the other hand, the set of clauses 8 *is not UP-redundant wrt to any encoding* that contains any of the other clauses (i.e. 1 to 7.3). From this we conclude that, for all problems P, BB-31$(P) >_{SUP}$ BB-7(P). Similarly clause sets 3 and 4 are not UP-redundant wrt to any other clause, and therefore, BB-31$(P) >_{SUP}$ SATPLAN06-4(P). Hence, it seems that from the implemented encodings of planning as satisfiability, BB-31 is the strongest. Finally, SATPLAN$^{max}(P) >_{SUP}$ BB-31(P), for any problem P, due to the existence of clause set 6.

Note, that the SATPLAN^{max} encoding uses only one set of mutex actions, namely set 7.1. However, it is possible that a clause is included in several sets of mutex clauses, each for a different reason. Therefore, a mutex pair of actions that belongs to set 7.1 may also belong to other mutex sets that are UP-redundant. The size of clause set 7.1 of SATPLAN^{max}, can be reduced by omitting all clauses of this set that also belong to sets 7.2 or 7.3. Furthermore, all mutex action clauses on actions A_1 and A_2 and time T that contain add effects p_1 and p_2 respectively such that p_1 and p_2 are mutex at time $T+1$ can also be omitted. We call the resulting encoding `SAT-MAX-PLAN`, or `SMP` for short.

Londex Propagation in Propositional Planning

From the STRIPS encoding P of a planning problem, we can construct its multi-valued representation P_M, using a translation M as those described e.g. in (Helmert 2009). For each multi-valued variable X of P_M with domain $\mathcal{D}_X$, we denote by $X(v)$ the fact in its STRIPS representation P that corresponds to the assignment of value $v \in \mathcal{D}_X$ to variable X. Moreover, $X(v,T)$ denotes the atom (in the planning graph and the propositional theory) that represents the truth value of $X(v)$ at time T. In order to abstract away from the details of the particular method that is used to construct the multi-valued representation of a STRIPS domain, and therefore simplify our discussion, we make some, we believe, natural assumptions about the domains we consider.

Definition 9 *A multi-valued translation method M that translates STRIPS problems into their multi-value representation, satisfies the* domain compatibility assumption *if for every STRIPS problem P and its multi-valued representation P_M the following conditions hold:*

1. *Let X be a multi-valued variable of P_M with domain $\mathcal{D}_X$ and A any action of P. If $X(v_i) \in add(A)$ for $v_i \in \mathcal{D}_X$, then $X(v_j) \in del(A) \cap pre(A)$ for some $v_j \in \mathcal{D}_X$ with $i \neq j$.*
2. *If X is a multi-valued variable of P_M with domain $\mathcal{D}_X$, then the initial state assigns true to exactly one fact of the form $X(v_i)$ for $v_i \in \mathcal{D}_X$.*

We can now prove that for translations that satisfy the domain compatibility assumption, Graphplan marks as mutex all facts that refer to the different values of a multi-valued variable.

Proposition 3 *Let P_M be the translation of a STRIPS problem P under a translation method M that satisfies the domain compatibility assumption. If X is a multi-valued variable of P_M with domain $\mathcal{D}_X$, in the planning graph all pairs of facts of the from $X(v_i), X(v_j)$, with $v_i, v_j \in \mathcal{D}_X$ and $i \neq j$, are mutex in all its levels where they both appear.*

Proof *We prove the claim inductively on planning graph levels.*
Base case. *Assume that both $X(v_i)$ and $X(v_j)$ appear on (fact) level 1 of the planning graph. We prove that they are marked as mutually exclusive by Graphplan. Suppose first that one of $X(v_i)$ and $X(v_j)$, say $X(v_i)$, appears in the initial state (fact level 0). Since $X(v_j)$ appears on level 1, there must be some actions $A_1^{v_j}, \dots A_n^{v_j}$ such that $X(v_j) \in add(A_c^{v_j})$ and $X(v_i) \in del(A_c^{v_j}) \cap pre(A_c^{v_j})$, for $1 \leq c \leq n$. On the other hand, $X(v_i)$ appears on level 1 because of $noopX(v_i)$. Observe that $noopX(v_i)$ is mutex with all actions $A_c^{v_j}$ in the preceding action level, and therefore $(X(v_i), X(v_j))$ are also mutex.*
Assume now that $X(v_t)$ appears in the initial state, and therefore, by the domain compatibility assumption, none of

$X(v_i)$ and $X(v_j)$ does. Then, there must be two sets of actions, $A_1^{v_j}, \ldots A_n^{v_j}$ and $A_1^{v_i}, \ldots A_m^{v_i}$, such that $X(v_j) \in add(A_c^{v_j})$, $X(v_t) \in del(A_c^{v_j}) \cap pre(A_c^{v_j})$, for $1 \leq c \leq n$, and $X(v_i) \in add(A_d^{v_i})$, $X(v_t) \in del(A_d^{v_i}) \cap pre(A_d^{v_i})$, for $1 \leq d \leq m$. Observe that every action $A_c^{v_j}$ deletes $X(v_t)$ which is a precondition of all actions $A_d^{v_i}$. Therefore, every action $A_c^{v_j}$ is mutex with every action $A_d^{v_i}$. Hence, $X(v_i)$ and $X(v_j)$ are also mutex at fact level 1.

Inductive hypothesis. *Assume that for some planning graph level k, Graphplan marks as mutex all pairs of facts of the form $X(v_i), X(v_j)$, with $v_i, v_j \in \mathcal{D}_X$.*

Inductive step. *We prove that the same holds for graph level $k+1$ for all pairs of facts of the from $X(v_i), X(v_j)$, with $v_i, v_j \in \mathcal{D}_X$. Let $A_c^{v_i}$ be an action such that $X(v_i) \in add(A_c^{v_i})$, and $A_d^{v_j}$ an action such that $X(v_j) \in add(A_d^{v_j})$. By the domain compatibility assumption, there are facts $X(v_c) \in del(A_c^{v_i}) \cap pre(A_c^{v_i})$, $X(v_d) \in del(A_d^{v_j}) \cap pre(A_d^{v_j})$, with $v_c, v_d \in \mathcal{D}_X$. Assume first that $v_c \neq v_d$. From the inductive hypothesis we know that $X(v_c), X(v_d)$ are mutex at fact level k, therefore actions $A_c^{v_i}$ and $A_d^{v_j}$ are mutex at action level k. On the other hand, if $v_c = v_d$, then $pre(A_c^{v_i}) \cap del(A_d^{v_j}) \neq \emptyset$, therefore $A_c^{v_i}$ and $A_d^{v_j}$ are again mutex. Therefore, any pair of actions $A_c^{v_i}$ and $A_d^{v_j}$ that add $X(v_i)$ and $X(v_j)$ respectively are mutex at level k. Hence, $X(v_i), X(v_j)$ are mutex at level $k+1$.* ■

A similar result can be proven (the proof is omitted) for actions that have multi-valued variables in their add effects.

Proposition 4 *Let P_M be the translation of a STRIPS problem P under a translation method M that satisfies the domain compatibility assumption. If X is a multi-valued variable of P_M with domain $\mathcal{D}_X$, in the planning graph all pairs of action A_i, A_j (including noops) such that $X(v_i) \in add(A_i)$, $X(v_j) \in add(A_j)$, for $v_i, v_j \in \mathcal{D}_X$, are mutex in all its levels where they both appear.*

In the rest of the paper we assume that londex constraints are generated from the multi-valued representation of a planning domain by a translation method that satisfies the domain compatibility assumption. Moreover, we assume that londex constraints are translated into clauses in a straightforward manner, i.e. a Class A action londex on actions A_1 and A_2 translates into a set of binary clauses $\neg A_1(T) \vee \neg A_2(T+1)$ for all valid points.

Londex propagation in SATPLAN06

In the following we analyze the effects of various londex constraints on the constraint propagation of a UP based SAT solver. A clause of the form $\neg p(T) \vee \neg q(T+k)$ that corresponds to a londex constraint of a planning problem P is *forward UP-redundant* wrt to an encoding T_P of P if $\neg q(T+k) \in UP(T_P \cup \{p(T)\})$. Similarly, the clause is *backward UP-redundant* wrt to T_P if $\neg p(T) \in UP(T_P \cup \{q(T+k)\})$.

We start by analyzing the effects of londex constraints of Class A. This class contains constraints that refer to actions that cannot be executed in parallel, as well as constraints that relate actions that are one time step apart. Note that action mutexes that refer to the same time point are included in `SATPLAN06` encoding, therefore we do not consider them. For the other type of Class A londex constraints, we show below that they are forward UP-redundant wrt the the SATPLAN06-4 model, which we refer to as `SATPLAN06` encodings.

Proposition 5 *Let A_1, A_2 be actions and f a fact of a planning problem P such that $f \in del(A_1)$ and $f \in pre(A_2)$. The set of clauses $\neg A_1(T) \vee \neg A_2(T+1)$, for all valid points T, is forward UP-redundant wrt the `SATPLAN06` encoding of the problem.*

Proof *Let T_P be the SATPLAN06 encoding of a problem P. We prove that $\neg A_2(T+1) \in UP(T_P \cup \{A_1(T)\})$. Let $A_1^f, A_2^f, \ldots, A_k^f$ be the actions that contain f in their add effects (including noop). Theory T_P contains the clauses*

1. $\neg A_2(T+1) \vee f(T+1)$
2. $\neg f(T+1) \vee A_1^f(T) \vee A_2^f(T) \vee \ldots \vee A_k^f(T)$
3. *A set of binary clauses of the form $\neg A_1(T) \vee \neg A_i^f(T)$, $1 \leq i \leq k$.*

From $A_1(T)$ and the set of clauses (3) above, UP derives the set of unit clauses $\neg A_i^f(T)$, $1 \leq i \leq k$ From these clauses and clause (2), UP entails the unit clause $\neg f(T+1)$ and from clause (1) $\neg A_2(T+1)$. Therefore $\neg A_2(T+1) \in UP(T \cup \{A_1(T)\})$.

■

We investigate now long distance constraints for facts, and show that they are also forward UP-redundant wrt to the `SATPLAN06` encoding.

Proposition 6 *Let X be a multi-valued variable of a planning problem P with domain $\mathcal{D}_X$, and T_P the SATPLAN06 encoding of P. Then, for any two values $v_i, v_j \in \mathcal{D}_X$ and $k \geq 0$ such that $\Delta_{G_X}(v_i, v_j) > k$, the set of clauses $\neg X(v_j, T+k) \vee \neg X(v_i, T)$ is forward UP-redundant wrt to T_P for all valid time points T.*

Proof *We prove inductively on k that $\neg X(v_j, T+k) \in UP(T_P \cup \{X(v_i, T)\})$ for all v_j s.t. $\Delta_{G_X}(v_i, v_j) > k$. Moreover, we show within the same inductive proof, that $\neg A^j(T+k) \in UP(T_P \cup \{X(v_i, T)\})$ for all actions A^j such that $X(v_j) \in pre(A^j)$ and $\Delta_{G_X}(v_i, v_j) > k$.*

Base case. *We prove that the theorem holds for $k = 0$, that is, if $\Delta_{G_X}(v_i, v_j) > 0$, $\neg X(v_j, T) \in UP(T_P \cup \{X(v_i, T)\})$. First note that $\Delta_{G_X}(v_i, v_j) > 0$ holds for all $v_i, v_j \in \mathcal{D}_X$, $j \neq i$. By proposition 3, T_P contains the clauses $\neg X(v_i, T) \vee \neg X(v_j, T)$, for all $v_i, v_j \in \mathcal{D}_X$, $j \neq i$. Therefore, $\neg X(v_j, T) \in UP(T_P \cup \{X(v_i, T)\})$. Furthermore, if A^j is an action such that $X(v_j) \in pre(A^j)$, then T_P contains the clause $\neg A^j(T) \vee X(v_j, T)$. From this clause and $\neg X(v_j, T) \in UP(T_P \cup \{X(v_i, T)\})$, we conclude that $\neg A^j(T) \in UP(T_P \cup \{X(v_i, T)\})$.*

Inductive hypothesis. *Assume that for some $k \geq 0$, $\neg X(v_j, T+k) \in UP(T_P \cup \{X(v_i, T)\})$ holds for all facts $X(v_j)$ such that $\Delta_{G_X}(v_i, v_j) > k$. Furthermore, $\neg A^j(T+k) \in UP(T_P \cup \{X(v_i, T)\})$ for all actions A^j such that $X(v_j) \in pre(A^j)$ and $\Delta_{G_X}(v_i, v_j) > k$.*

Inductive step. *We prove first that $\neg X(v_j, T+k+1) \in UP(T_P \cup \{X(v_i, T)\})$ holds for all facts $X(v_j)$ such that*

$\Delta_{G_X}(v_i, v_j) > k+1$. *Let* $A_1^j, A_2^j, \ldots, A_m^j$ *be the actions that have* $X(v_j)$ *in their add effects. Then* T_P *contains the clause*
$\neg X(v_j, T+k+1) \vee A_1^j(T+k) \vee A_2^j(T+k) \vee \ldots \vee A_m^j(T+k) \vee noopX(v_j, T+k)$.
Since $\Delta_{G_X}(v_i, v_j) > k+1$, *implies* $\Delta_{G_X}(v_i, v_j) > k$, *by the inductive hypothesis* $\neg X(v_j, T+k) \in UP(T_P \cup \{X(v_i, T)\})$. *From this and the binary clause* $\neg noopX(v_j, T+k) \vee X(v_j, T+k)$ *we conclude* $\neg noopX(v_j, T+k) \in UP(T_P \cup \{X(v_i, T)\})$. *Assume now that there is some action* A_c^j, *for* $1 \leq c \leq m$, *such that* $\neg A_c^j(T+k) \notin UP(T_P \cup \{X(v_i, T)\})$, *and let* $X(v_b)$, $v_b \in \mathcal{D}_X$, *be a precondition of* A_c^j. *Then, it can not be the case that* $\Delta_{G_X}(v_i, v_b) > k$, *because then, by the induction hypothesis,* $\neg A_c^j(T+k) \in UP(T_P \cup \{X(v_i, T)\})$. *Therefore,* $\Delta_{G_X}(v_i, v_b) \leq k$. *Then there must exist a path in* G_X *from* v_i *to* v_b *of length at most* k, *and a arc from* v_b *to* v_j, *therefore* $\Delta_{G_X}(v_i, v_j) \leq k+1$. *However, this contradicts the assumption* $\Delta_{G_X}(v_i, v_j) > k+1$. *Therefore, it must be the case that* $\neg A_c^j(T+k) \in UP(T_P \cup \{X(v_i, T)\})$, *for all* $1 \leq c \leq m$. *Hence,* $\neg X(v_j, T+k+1) \in UP(T_P \cup \{X(v_i, T)\})$.

We now prove that $\neg A^j(T+k+1) \in UP(T_P \cup \{X(v_i, T)\})$ *for all actions* A^j *such that* $X(v_j) \in pre(A^j)$ *and* $\Delta_{G_X}(v_i, v_j) > k+1$. *From the first part of the proof we know that* $\neg X(v_j, T+k+1) \in UP(T_P \cup \{X(v_i, T)\})$. *Moreover, theory* T_P *contains the clause* $\neg A^j(T+k+1) \vee X(v_j, T+k+1)$. *Therefore,* $\neg A^j(T+k+1) \in UP(T_P \cup \{X(v_i, T)\})$. *This completes the proof.* ■

The results that follow show that all forms of Class B action londex constraints are forward UP-redundant. The proofs of these propositions give some insight into the propagation taking place in a UP-based SAT solver, and are provided only for two of them.

Proposition 7 *Let* X *be a multi-valued variable of a planning problem* P *with domain* $\mathcal{D}_X$, T_P *the SATPLAN06 encoding of* P, *and* $v_1, v_2 \in \mathcal{D}_X$ *such that* $\Delta_{G_X}(v_1, v_2) > k$. *If* A_1, A_2 *are actions such that* $X(v_1) \in pre(A_1)$ *and* $X(v_2) \in pre(A_2)$, *then the set of clauses* $\neg A_2(T+k) \vee \neg A_1(T)$ *is forward UP-redundant wrt to* T_P *for all valid time points.*

Proof *We show that* $\neg A_2(T+k) \in UP(T_P \cup \{A_1(T)\})$. *Theory* T_P *contains the clauses* $\neg A_1(T) \vee X(v_1, T)$ *and* $\neg A_2(T+k) \vee X(v_2, T+k)$. *Therefore,* $X(v_1, T) \in UP(T_P \cup \{A_1(T)\})$. *By proposition 6,* $\neg X(v_2, T+k) \in UP(T_P \cup \{X(v_1, T)\}$, *and therefore* $\neg X(v_2, T+k) \in UP(T_P \cup \{A_1(T)\})$. *By the clause* $\neg A_2(T+k) \vee X(v_2, T+k)$, *we obtain* $\neg A_2(T+k) \in UP(T_P \cup \{A_1(T)\})$. ■

Proposition 8 *Let* X *be a multi-valued variable of a planning problem* P *with domain* $\mathcal{D}_X$, T_P *the SATPLAN06 encoding of* P, *and* $v_1, v_2 \in \mathcal{D}_X$ *such that* $\Delta_{G_X}(v_1, v_2) > k$. *If* A_1, A_2 *are actions such that* $X(v_1) \in add(A_1)$ *and* $X(v_2) \in add(A_2)$, *then the set of clauses* $\neg A_2(T+k) \vee \neg A_1(T)$ *is forward UP-redundant wrt to* T_P *for all valid time points.*

Proof *We prove inductively on* k *that* $\neg A_2(T+k) \in UP(T_P \cup \{A_1(T)\})$ *for any pair of actions* A_1, A_2 *such that* $X(v_1) \in add(A_1)$ *and* $X(v_2) \in add(A_2)$ *and* $\Delta_{G_X}(v_1, v_2) > k$.

Base case: *We prove first the case* $k = 0$. *Note that* $\Delta_{G_X}(v_1, v_2) > 0$ *for all* $v_1, v_2 \in \mathcal{D}_X$, $j \neq i$. *Therefore, we must show that* $\neg A_2(T) \in UP(T_P \cup \{A_1(T)\})$ *for any pair of actions* A_1, A_2 *such that* $X(v_1) \in add(A_1)$ *and* $X(v_2) \in add(A_2)$ *with* $v_1 \neq v_2$. *Assume first that* $X(v_p) \in pre(A_1) \cap pre(A_2)$ *for* $v_p \in \mathcal{D}_X$. *Then* $X(v_p) \in del(A_1) \cap del(A_2)$, *therefore* $\neg A_2(T) \vee \neg A_1(T) \in T_P$. *Assume now that* $X(v_p^1) \in pre(A_1)$ *and* $X(v_p^2) \in pre(A_2)$ *with* $v_p^1, v_p^2 \in \mathcal{D}_X$ *and* $v_p^1 \neq v_p^2$. *Theory* T_P *contains the clauses* $\neg A_2(T) \vee X(v_p^2, T)$ *and* $\neg A_1(T) \vee X(v_p^1, T)$. *By proposition 3, it also contains the clause* $\neg X(v_p^1, T) \vee \neg X(v_p^2, T)$. *From these clauses it follows that* $\neg A_2(T) \in UP(T_P \cup \{A_1(T)\})$.

Inductive hypothesis. *Assume that for some* $k \geq 0$, $\neg A_2(T+k) \in UP(T_P \cup \{A_1(T)\})$ *holds for all pairs of actions* A_1, A_2 *such that* $X(v_1) \in add(A_1)$ *and* $X(v_2) \in add(A_2)$ *and* $\Delta_{G_X}(v_1, v_2) > k$.

Inductive step: *We prove that* $\neg A_2(T+k+1) \in UP(T_P \cup \{A_1(T)\})$ *holds for all pairs of actions* A_1, A_2 *such that* $X(v_1) \in add(A_1)$ *and* $X(v_2) \in add(A_2)$ *and* $\Delta_{G_X}(v_1, v_2) > k+1$. *Let* $v_p^2 \in pre(A_2)$ *with* $v_p^2 \in \mathcal{D}_X$, *and let* $A_2^{p_c}$, $1 \leq c \leq n$, *be the set of actions that have* $X(v_p^2)$ *in their add effects. Clearly,* $\Delta_{G_X}(v_1, v_p^2) > k$. *From the inductive hypothesis we know that* $\neg A_2^{p_c}(T+k) \in UP(T_P \cup \{A_1(T)\})$ *for all* $1 \leq c \leq n$. *Theory* T_P *contains the clause* $\neg X(v_p^2, T+k+1) \vee A_2^{p_1}(T+k) \vee \ldots \vee A_2^{p_n}(T+k) \vee noop(v_p^2, T+k)$.
We prove now that $\neg noop(v_p^2, T+k) \in UP(T_P \cup \{A_1(T)\})$. *Let* $v_p^1 \in pre(A_1)$ *with* $v_p^1 \in \mathcal{D}_X$ *and assume first that* $v_p^1 \neq v_p^2$. *First note that* $X(v_p^1, T) \in UP(T_P \cup \{A_1(T)\})$ *and assume that* $\neg X(v_p^2, T+k) \notin UP(T_P \cup \{A_1(T)\})$, *therefore* $\neg X(v_p^2, T+k) \notin UP(T_P \cup \{X(v_p^1, T)\})$. *By proposition 3 this means that* $\Delta_{G_X}(v_p^1, v_p^2) \leq k$, *which implies* $\Delta_{G_X}(v_1, v_2) \leq k$, *contradicting the assumption that* $\Delta_{G_X}(v_1, v_2) > k+1$. *Hence,* $\neg X(v_p^2, T+k) \in UP(T_P \cup \{A_1(T)\})$. *From this and the clause* $\neg noop(v_p^2, T+k) \vee X(v_p^2, T+k)$ *we obtain* $\neg noop(v_p^2, T+k) \in UP(T_P \cup \{A_1(T)\})$. *Similar arguments hold in the case* $v_p^1 = v_p^2$.
From the above analysis we conclude that $\neg X(v_p^2, T+k+1) \in UP(T_P \cup \{A_1(T)\})$. *From this, and the clause* $\neg A_2(T+k+1) \vee X(v_p^2, T+k+1)$ *we obtain* $\neg A_2(T+k+1) \in UP(T_P \cup \{A_1(T)\})$. ■

Analogous results can be proven for the other two types of Class B londex constraints.

Proposition 9 *Let* X *be a multi-valued variable of a planning problem* P *with domain* $\mathcal{D}_X$, T_P *the SATPLAN06 encoding of* P, *and* $v_1, v_2 \in \mathcal{D}_X$ *such that* $\Delta_{G_X}(v_1, v_2) > k$. *Then, the following sets of clauses are forward UP-redundant wrt to* T_P *for all valid time points.*

1. $\neg A_2(T+k) \vee \neg A_1(T-1)$, *for* A_1, A_2 *such that* $X(v_1) \in add(A_1)$ *and* $X(v_2) \in pre(A_2)$.

2. *$\neg A_2(T+k-1) \vee \neg A_1(T)$, for A_1, A_2 such that $X(v_1) \in pre(A_1)$ and $X(v_2) \in add(A_2)$.*

Since BB-31 $\geq_{SUP}$ SATPLAN06, the same results hold for the BLACKBOX encoding. However, it can be shown that BLACKBOX encoding can not propagate backwards some of the mutex constraints (details regarding this issue can be found in an extended version of the paper). Therefore, londex constraints increase propagation in both SATPLAN06 and BLACKBOX encodings.

Londex Propagation in SATPLAN^{max}

In this section we prove that in the SATPLAN^{max} encoding all londex constraints are UP-redundant in both directions, forward and backwards.

It was proved earlier that (clauses that correspond to) londexes are forward UP-redundant in the SATPLAN06 encoding. Since $SATPLAN^{max} >_{SUP} SATPLAN06$, londexes are forward UP-redundant in SATPLAN^{max} as well.

Proposition 10 *Let X be a multi-valued variable of a planning problem P with domain $\mathcal{D}_X$, T_P the $SATPLAN^{max}$ encoding of P, and $v_1, v_2 \in \mathcal{D}_X$ such that $\Delta_{G_X}(v_1, v_2) > k$. The set of clauses $\neg X(v_2, T+k) \vee \neg X(v_1, T)$ is backward UP-redundant wrt to T_P, for all valid time points.*

Proof *We prove inductively on k that $\neg X(v_1, T) \in UP(T_P \cup \{X(v_2, T+k)\})$.*

Base case. *For $k = 0$, it follows form proposition 3 that $X(v_1)$ and $X(v_2)$ are marked mutually exclusive on all planning graph levels, therefore T_P contains the clause $\neg X(v_2, T) \vee \neg X(v_1, T)$. Hence $\neg X(v_1, T) \in UP(T_P \cup \{X(v_2, T)\})$.*

Inductive hypothesis. *Assume that for any pair of facts $X(v_1), X(v_2)$ and some $k \geq 0$ with $\Delta_{G_X}(v_1, v_2) > k$, it holds that $\neg X(v_1, T) \in UP(T_P \cup \{X(v_2, T+k)\})$.*

Inductive step. *We show that for any pair of facts $X(v_1), X(v_2)$ with $\Delta_{G_X}(v_1, v_2) > k+1$, it holds that $\neg X(v_1, T) \in UP(T_P \cup \{X(v_2, T+k+1)\})$.*
By the definition of the DTG, in the $G_X = (V, E)$ containing variables $X(v_1)$ and $X(v_2)$ with $\Delta_{G_X}(v_1, v_2) > k+1$, there exist (other) variables $X(v_{21}), X(v_{22}), \ldots, X(v_{2n})$ such that $\{(X(v_1), X(v_{21})), \ldots, (X(v_1), X(v_{2n}))\} \subset E$, $\Delta_{G_X}(v_1, v_i) = 1$ and $\Delta_{G_X}(v_i, v_2) > k$ for $X(v_i) \in \{X(v_{21}), X(v_{22}), \ldots, X(v_{2n})\}$. For any variable $X(v_i) \in \{X(v_{21}), X(v_{22}), \ldots, X(v_{2n})\}$, there is an associated set of actions $\{A_1^{v_i}, A_2^{v_i}, \ldots, A_{m_{v_i}}^{v_i}\}$ each one having $X(v_i)$ as an add effect, and $X(v_1)$ as a precondition and delete effect. It holds that $\{\neg A_1^{v_{21}}(T) \vee X(v_{21}, T+1), \ldots, \neg A_{m_{v_{21}}}^{v_{21}}(T) \vee X(v_{21}, T+1), \ldots, \neg A_1^{v_{2n}}(T) \vee X(v_{2n}, T+1), \ldots, \neg A_{m_{v_{2n}}}^{v_{2n}}(T) \vee X(v_{2n}, T+1)\} \subset T_P$ and $\{\neg X(v_1, T) \vee X(v_1, T+1) \vee A_1^{v_{21}}(T) \vee \ldots \vee A_{m_{v_{21}}}^{v_{21}}(T) \vee \ldots \vee A_1^{v_{2n}}(T) \vee \ldots \vee A_{m_{v_{2n}}}^{v_{2n}}(T)\} \subset T_P$. Since $\forall X(v_i) \in \{X(v_{21}), X(v_{22}), \ldots, X(v_{2n})\}$ holds that $\Delta_{G_X}(v_i, v_2) > k$ by the inductive hypothesis it holds that $\{\neg X(v_{21}, T+1), \neg X(v_{22}, T+1), \ldots, \neg X(v_{2n}, T+1)\} \subseteq UP(T_P \cup \{X(v_2, T+k+1)\})$, which are further (unit) resolved with the binary clauses $\neg A_1^{v_{21}}(T) \vee X(v_{21}, T+1), \ldots, \neg A_{m_{v_{21}}}^{v_{21}}(T) \vee X(v_{21}, T+1), \ldots, \neg A_1^{v_{2n}}(T) \vee X(v_{2n}, T+1), \ldots, \neg A_{m_{v_{2n}}}^{v_{2n}}(T) \vee X(v_{2n}, T+1)$, giving $\{\neg A_1^{v_{21}}(T), \ldots, \neg A_{m_{v_{2n}}}^{v_{2n}}(T)\} \subseteq UP(T_P \cup \{X(v_2, T+k+1)\})$. Hence the clause $\neg X(v_1, T) \vee X(v_1, T+1) \vee A_1^{v_{21}}(T) \vee \ldots \vee A_{m_{v_{21}}}^{v_{21}}(T) \vee \ldots \vee A_1^{v_{2n}}(T) \vee \ldots \vee A_{m_{v_{2n}}}^{v_{2n}}(T)$ is resolved to the binary clause $\neg X(v_1, T) \vee X(v_1, T+1)$. But because $\Delta_{G_X}(v_1, v_2) > k$ (since $\Delta_{G_X}(v_1, v_2) > k+1$), by the inductive hypothesis it holds that $\neg X(v_1, T+1) \in UP(T_P \cup \{X(v_2, T+k+1)\})$, which further resolves the binary clause $\neg X(v_1, T) \vee X(v_1, T+1)$ to $\neg X(v_1, T)$. ■

Based on proposition 10, we can prove that all six categories of action londexes are backward UP-redundant. By combining the results of this and the previous section, we obtain the following property for the SATPLAN^{max} encoding.

Theorem 1 *Let P be STRIPS planning domain and T_P its SATPLAN^{max} encoding. All clauses that correspond to londex constraints derived from P are UP-redundant wrt T_P.*

Note that the above result holds for SMP as well, as it is a simplification of SATPLAN^{max} obtained by removing UP-redundant clauses.

Experimental evaluation

In this section we present the results of the experimental comparison of various encodings discussed earlier, in domains from the last planning competitions. Our implementation is an extension of the SATPLAN06 system with new encodings for BLACKBOX and SMP as well as the integration of precosat. Hence, all experiments are runs of the same system with different values for the parameters *encoding* and *solver*. The experiments were run on an IBM X3650 with Intel Xeon processors at 2.0 GHz and 32GB of RAM, running under CentOS 5.2.

Table 1 presents the number of problems solved with different combinations of encodings and SAT solvers, within a CPU time limit of 2500 seconds. The encodings compared are SATPLAN06 (encoding SATPLAN06-4), BLACKBOX (encoding BB-31), and SMP. The SAT solvers that are used are siege (Ryan 2003) and precosat (Biere 2009) version 236, a new system that seems to outperform Siege and many other solvers that we have tested on a large number of planning domains. In Table 1 (as well as Table 2) SATPLAN06 is denoted by SP, BLACKBOX by BB, whereas siege by SI and precosat by PR.

The entries under "Problems" in Table 1 are of the form p/q, where p is the total number of problems contained in each domain, and q the number of problems for which either one of the methods found a solution, or they all reached their CPU limit and terminated without a solution. Hence, $p - q$ is the number of problems that were not solved by any of the systems due to memory problems at parsing or solving time.

Table 2 presents characteristic run times of different encodings on some of the hardest problems that have been solved by both BLACKBOX and SMP. All times were obtained with precosat as the underlying SAT solver, and a CPU time limit of 3600 seconds.

Domain	Problems	SP-SI	BB-SI	SMP-SI	SP-PR	BB-PR	SMP-PR
Depots	22/22	16	16	18	17	17	19
DriveLog	20/20	16	16	16	17	17	17
Zenotravel	20/19	15	15	16	15	15	16
Freecell	20/20	4	4	6	5	5	6
Satellite	36/24	17	17	17	17	18	18
Pathways	30/30	9	9	10	12	12	16
Trucks	30/30	5	6	8	7	7	11
Pipes	50/31	17	23	23	15	24	27
Storage	30/30	15	15	15	15	15	16
TPP	30/30	27	28	28	28	29	29
Elevators	30/30	9	9	12	12	13	14
ScanAnalyser	30/23	17	17	19	15	16	18
Sokoban	30/30	2	2	4	2	5	7
Transport	30/21	11	11	12	11	11	13
Total	408/360	180	188	204	188	204	227

Table 1: Number of problems solved by each encoding in different domains.

Domain-Problem	SMP	BB	SP
Depots-11	176	1674	2134
DriveLog-16	897	1156	2453
Zenotravel-15	84	307	383
Pathways-17	971	980	1940
Trucks-8	161	637	1140
TPP-21	1580	1908	2554
Pipes-12	189	348	1429
Transport-4	81	312	563
Sokoban-13	474	1869	-
Elevator-21	2099	2424	-
ScanAnalyser-8	59	208	-

Table 2: Run times in seconds for different encodings of problems. A dash indicates CPU timeout.

The relative performance of the different encodings, as depicted in Tables 1 and 2, is consistent with the theoretical results obtained in earlier sections. Indeed, BLACKBOX outperforms SATPLAN06, whereas SMP dominates all other encodings. Moreover, solution times improve when precosat instead of siege is used as the SAT solver.

Conclusions

In this work we compared different encodings of planning as satisfiability wrt to the propagation they achieve in a modern SAT solver. Our theoretical results explain some of the differences observed in the performance of various planners. One interesting finding is that BLACKBOX encoding is stronger than the one of SATPLAN06. Thus, new encodings of planning as satisfiability need to be compared with both systems. Another practical outcome of our results is SMP, a new encoding that renders londex constraint redundant, and seems to offer performance improvements in a number of domains.

Our ongoing work includes the study of action based encodings, the new londex constraints of (Chen et al. 2009), and the propagation in the split action model of (Robinson et al. 2009).

References

Biere, A. 2009. P{re,ic}oSAT@SC'09. In *SAT Competition 2009*, http://fmv.jku.at/precosat/.

Blum, A., and Furst, M. L. 1997. Fast planning through planning graph analysis. *Artif. Intell.* 90(1-2):281–300.

Chen, Y.; Huang, R.; Xing, Z.; and Zhang, W. 2009. Long-distance mutual exclusion for planning. *Artif. Intell.* 173(2).

Chen, Y.; Xing, Z.; and Zhang, W. 2007. Long-distance mutual exclusion for propositional planning. In *IJCAI*, 1840–1845.

Een, N., and Sorensson, N. 2003. An extensible sat-solver. In *SAT 2003*.

Geffner, H. 2004. Planning graphs and knowledge compilation. In *ICAPS*, 52–62.

Helmert, M. 2009. Concise finite-domain representations for PDDL planning tasks. *Artif. Intell.* 173(5-6):503–535.

Kautz, H. A., and Selman, B. 1999. Unifying sat-based and graph-based planning. In *IJCAI*, 318–325.

Kautz, H. A.; Selman, B.; and Hoffmann, J. 2006. SATPLAN: Planning as satisfiability. In *Booklet of the 5th Planning Competition*.

Rintanen, J. 2008. Planning graphs and propositional clause-learning. In *KR*, 535–543.

Robinson, N.; Gretton, C.; Pham, D. N.; and Sattar, A. 2009. SAT-based parallel planning using a split representation of actions. In *ICAPS*.

Ryan, L. 2003. Efficient algorithms for clause-learning sat solvers. In *M.Sc. Thesis, Simon Fraser University*.