Twenty-Eighth International Conference on Automated Planning and Scheduling (ICAPS 2018)

A TIL-Relaxed Heuristic for Planning with Time Windows

Tony Allard
Cyber and Electronic Warfare Division
Defence Science and Technology Group
firstname.lastname@dst.defence.gov.au

Charles Gretton
Australian National University
firstname.lastname@anu.edu.au

Patrik Haslum
Australian National University
and CSIRO Data61
firstname.lastname@anu.edu.au

Abstract

We consider planning problems with *time windows*, in which the availability of discrete resources is time constrained. We develop a novel heuristic that addresses specifically the difficulty of coordinating actions within time windows. The heuristic is based on solving a temporally relaxed problem and measuring the magnitude by which the relaxed solution violates the time window constraints. Applied in a state-space search planner, the heuristic reduces the number of dead-ends encountered during search, and improves planner coverage.

Introduction

Many interesting real-world problems require action within a prescribed time window. For example, a vehicle routing problem (VRP) where cargoes must be delivered before a shop closes (Kolen, Rinnooy Kan, and Trienekens 1987), a satellite transmitting data to earth while over a receiving ground station (Globus et al. 2004), or the transportation of oil derivatives through a pipeline network within deadlines (Milidiú, dos Santos Liporace, and de Lucena 2003). Like Gerevini, Saetti, and Serina (2005) we focus on Planning Problems with Time Windows (PPTW) where time windows are determined by exogenous events that happen at known times. Coles et al. (2008) note that a core planning difficulty in this setting occurs where multiple actions require exclusive use of a limited resource, and must coordinate to share that resource during periods of availability.

For many PPTW the underlying causal problem—i.e., the timeless problem of choosing a series of actions that logically achieve the goal—is easy. We caution that in general, a causal problem in itself may of course be difficult. If the causal problem is easy, then the difficulty of PPTWs follows from the interaction between the causal and temporal constraints. For example, the small delivery problem shown in Figure 2 requires goods to be carried first by a truck and then by an aircraft, with several options for where to hand over from one to the other. Causally, these options are all equivalent, but the imposition of time windows can render some of them infeasible. Recognising the implications that time windows have on the causal structure of the plan is key to solving PPTWs. Existing planners are not good at eliciting such knowledge in the early stages of search, and therefore often explore many causal decisions which are temporally infeasible. Late recognition of such dead-end states causes computationally expensive search backtracking. This is illustrated by the experiment result in Figure 1. Here, we have taken a state-of-the-art temporal planner, POPF (Coles et al. 2010), and applied it to sets of instances of the multi-modal cargo routing (MMCR) domain (Allard and Gretton 2015) with time windows varying in tightness. We compare the time it takes POPF to solve these problems with the time it takes to solve a relaxation of the same instances, in which we enforce only the logical constraints of the time windows (i.e., their occurrance and order) but ignore the quantitative time constraints. We call this the "TIL-relaxation" of the problem. We do *not* relax any of the causal constraints on plans. The experiment shows that the tighter time windows become, the harder the problems are. The hardness of the relaxed problem, however, remains unaffected.

Temporal planners based on forward state-space search, such as POPF (Coles et al. 2010) and COLIN (Coles et al. 2012), draw on heuristics that have been developed to solve the classical, causal, planning problem, and that do so very effectively. While the heuristics used by these temporal planners have some adaptation to the temporal planning setting, this experiment shows they are still not able to anticipate time window violations casused by poor causal decisions, leading to time-consuming search, and, in many cases, failure to solve problems with tight time windows.

Based on this insight, we propose a novel heuristic for PPTWs that targets the specific difficulty caused by tight time windows. This heuristic is based on solving the TIL-relaxed problem—i.e, relaxing the quantitative time window constraints, but preserving all causal constraints—and measuring the magnitude to which the solution to the relaxed problem violates the time windows of the original problem. We have implemented this heuristic within two forward state-space search-based temporal planners, COLIN and POPF, in each case using the same planner to solve the TIL-relaxed problem. Although this is computationally demanding—we compile and solve a complete planning problem in every heuristic evaluation—we demonstrate that the search guidance it provides is superior in several domains, leading to fewer dead-end states encountered and fewer states evaluated overall, compared to each planner

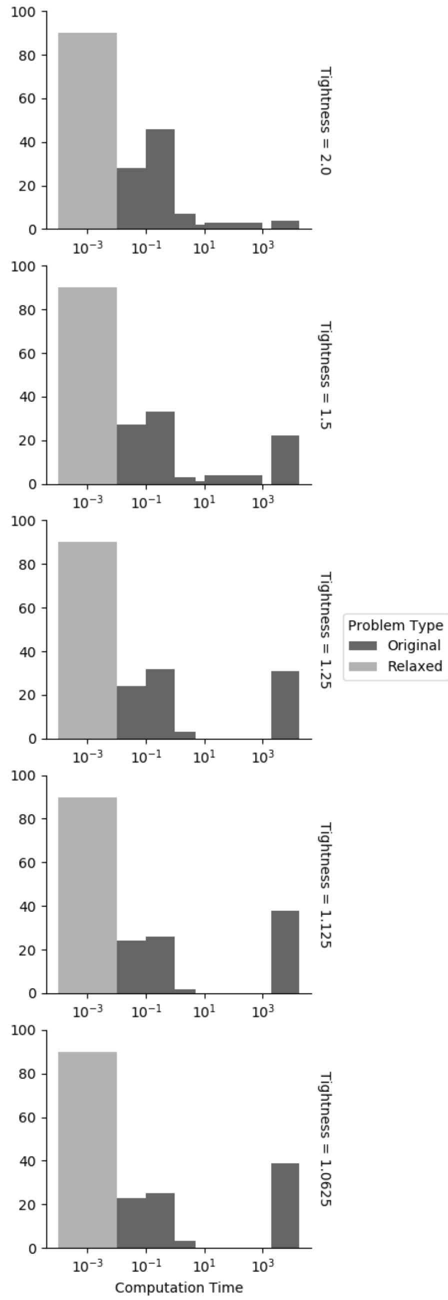

Figure 1: Distribution of POPF solution times for original and TIL-relaxed instances of the MMCR domain with varying time window tightness. The instances with the loosest time windows are at the top (tightness $= 2$) and the instances with the tightest time windows are at the bottom (tightness $= 1.0625$). An upper limit of 30 minutes was set for computation time. The column furthest right shows the number of problem instances for which a plan could not be found within the time limit.

with its original heuristic. When the TIL-relaxed problem is easy to solve (as shown in the example of the MMCR domain), it is enough to translate into a reduction in the overall runtime and an increase in instances solved.

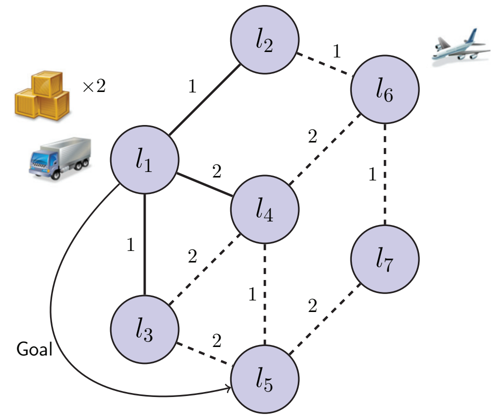

Figure 2: Example MMCR Scenario. Solid and dashed lines indicate road and air routes respectively. The directed arc shows the goal is to deliver two units of cargo from location l_1 to l_5.

A Motivating Example

An exemplar scenario for PPTWs is the multi-modal cargo routing (MMCR) domain (Allard and Gretton 2015), a challenging variety of vehicle routing problem. The MMCR domain defines cargoes, where the goal is to have those delivered within specific time windows. This is a rich heterogeneous problem in which different vehicles may be capable of reaching distinct sub-sets of locations. Vehicle movement and loading/unloading of cargo are actions which execute over time. Cargo can only be interacted with (loaded, unloaded and moved) during specific time windows defined for each problem. In a plan, vehicles may be required to exchange cargo at known locations in order to complete delivery. Time windows can often be tight.

An instance of the MMCR domain is depicted in Figure 2. Nodes represent locations within a transportation network. Solid and dashed edges linking nodes represent road and air routes between the corresponding locations, respectively. Edges are labelled with travel times. The goal is to have the two units of cargo, initially located at l_1, delivered to l_5. That goal can only be achieved using both the truck and aircraft, initially located at l_1 and l_6, respectively. Each vehicle requires 1 unit of time to load/unload a unit of cargo at each location. A time window constrains any interaction with cargo to be between the hours of 1 and 12. Generally the MMCR domain has capacitated vehicles and locations. To keep this running example simple, we place no such restrictions on either locations or vehicles.

Taking a closer look at the problem it becomes apparent that for delivery to be successful, a handover must occur between the truck and aircraft. This can take place at any of the locations l_2, l_3, or l_4. While all choices provide a causally valid plan, due to time window constraints only l_3 and l_4 provide a solution that is both casually valid and consistent with respect to time window constraints. For both these plans, the schedule of actions interacting with the cargo fits exactly within the time window duration. This is the tightest a time window could be and still allow for a plan. Investigations have shown that planning becomes increasingly more difficult as tightness increases (Gent et al. 1996).

Problem Description

We adopt the following definition of a PPTW, which is compatible with the PDDL 2.2 semantics (Edelkamp and Hoffmann 2004) and with the models used in previous work (Gerevini, Saetti, and Serina 2005; Coles et al. 2008). A PPTW is a quintuple $P = \langle V, A, T, I, G \rangle$, where:

1. V is the set of propositions, such that a complete truth assignment describes a state, s,
2. A is the set of all actions (instantaneous and durative),
3. T is a series of exogenous events, e, of the form $e = \langle t, L \rangle$; where L specifies the literals(s) to be realised when e occurs at a time step t,
4. I is a complete truth assignment corresponding to the initial state, and
5. G is a (partial) truth assignment which describes goal states. State s is a goal state iff $G \subseteq s$.[1]

A durative action a is defined by a tuple $\langle \mathit{Scond}, \mathit{Ocond}, \mathit{Econd}, \mathit{Seff}, \mathit{Eeff}, \mathit{dur} \rangle$, whose elements are the starting, ending and overall conditions, the starting and ending effects, and the action's duration. Durative action semantics are given according to a closed interval, $[t_s, t_e]$, the action is scheduled to execute in. The duration, $\mathrm{dur}(a)$, is a positive quantity, and the equality $t_e - t_s = \mathrm{dur}(a)$ must be satisfied. The state s_s immediately prior to time t_s must satisfy the starting conditions, i.e., $\mathit{Scond}(a) \subseteq s_s$. The starting effect takes place at t_s, producing a new state s'_s at t_s such that $\mathit{Seff}(a) \subseteq s'_s$. As usual, the values of variables not changed by the action effects persist. Likewise, the state s_e immediately before t_e must satisfy $\mathit{Econd}(a) \subseteq s_e$, and the ending of the action produces a new state s'_e at t_e which satisfies $\mathit{Eeff}(a) \subseteq s'_e$. Any state s_o that occurrs during the open interval (t_s, t_e) must satisfy $\mathit{Ocond}(a) \subseteq s_o$. We also allow for instantaneous actions where $0 = \mathrm{dur}(a)$ and $\emptyset = \mathit{Seff}(a) = \mathit{Econd}(a) = \mathit{Ocond}(a)$. Given a scheduled interval $[t_s, t_e]$, the action effects, $\mathit{Seff}(a)$ and $\mathit{Eeff}(a)$, can be modelled as events $\langle t_s, \mathit{Seff}(a) \rangle$ and $\langle t_e, \mathit{Eeff}(a) \rangle$, respectively. Several events (instantaneous actions, durative action starts or ends, or TILs) can occur at the same instant, provided they are commutative. As per PDDL 2.2 semantics, occurrence times of non-commutative events must be separated by some minimum time, which we denote with ϵ.

An event $e = \langle t, L \rangle$ affects the truth values of variables associated with L so that the state s_t at time t is consistent with L. When two exogenous events constrain the availability of a resource, r, at times $t_s < t_e$, we term the relationship between the events a time window. In PDDL 2.2 (Edelkamp and Hoffmann 2004) these events are modelled as Timed Initial Literals (TILs). Here, r becomes available at t_s, and is made unavailable at t_e. To model a time window constrained resource r, we have pairs of events $\langle t_s, \mathrm{avail}(r) \rangle$ and $\langle t_e, \neg\, \mathrm{avail}(r) \rangle$ in T.

[1] As usual, we write $G \subseteq s$ to mean all literals in G hold in s.

Algorithm 1 The TIL-Relaxed Heuristic

Input: a state s, a plan prefix Γ, PPTW P
Output: $h(s)$, heuristic value for s
1 $\hat{s}, \hat{P} \leftarrow$ TILRELAXATION(s, P)
2 $\hat{\pi} \leftarrow \Gamma \cup$ SOLVE$(\hat{s}, \hat{P})$
3 $\hat{\xi} \leftarrow$ DEORDERKK$(\hat{\pi})$
4 $\hat{G} \leftarrow$ BUILDSTN$(\hat{\pi}, \hat{\xi})$
5 $G \leftarrow$ TIGHTENTILCONSTRAINTS$(\hat{G}, T)$
6 $C = \emptyset$
7 **repeat**
8 $\quad C \leftarrow C \cup$ EXTRACTCONFLICTS(G)
9 $\quad h(s) \leftarrow$ MINIMUMTEMPORALRELAXATIONS(C)
10 **until** G is consistent
11 **return** $h(s)$

The solution to a PPTW is a temporally flexible plan, $\pi = \langle \alpha, \xi \rangle$. Here, $\alpha = \langle e_1, \ldots, e_n \rangle$ is the set of instances of actions and exogenous events that describes a valid causal transition from I to G. ξ is a set of temporal constraints between events in α. Each temporal constraint is of the form $e_i - e_j \in [l, u]$, i.e., bounds the difference in time between two events. ξ can be represented as a Simple Temporal Network (STN) (Dechter, Meiri, and Pearl 1991). Figure 3 shows an example of a temporal plan as an STN. Vertices are events, and edges temporal constraints between them. For example, events e_1 and e_3 are the start and end of an action, respectively, and the constraint $[2, 2]$ between them is the action's duration. Event e_{14} is the start of another action, which is causally dependent on the action ending at e_3; the constraint $e_{14} - e_3 \geq \epsilon$ expresses the precedence constraint between the actions.

The TIL-Relaxed Heuristic (TRH)

To solve PPTW we introduce a new heuristic which computes a plan for a TIL-relaxed version of the problem. In that plan the occurrence of TIL events is relatively unconstrained. A plan for the TIL-relaxed problem may be invalid for the concrete PPTW. The TRH value is the magnitude that events must be shifted so that the TIL-relaxed plan is valid.

Algorithm 1 shows the TRH algorithm as applied to a state, s, during planning. The inputs are the PPTW, P, the state evaluated, s, and the actions taken from the initial state to reach s (the plan prefix), Γ. The algorithm produces a relaxed problem by transforming TIL events into instantaneous actions. This step is described in detail in the next section. The result is a TIL-relaxed problem $\langle \hat{s}, \hat{P} \rangle$ (line 1). A solution to the relaxed problem is found using a general planner (line 2). This solution is concatenated with the prefix Γ to produce a relaxed plan, $\hat{\pi}$. A deordering of the relaxed plan is then determined and temporal constraints between events used to construct an STN, $\hat{G}$ (lines 3-4). Temporal constraints on TILs within the STN are tightened to their original values from P (line 5), to produce a tightened network, G. Any constraints that violate the consistency of the network are identified and used to determine the magnitude of violation (lines 6-10). This process is described more detail below. The magnitude of violation is returned as a heuristic value, $h(s)$, of the state (line 11).

Defining the TIL-Relaxed Problem: $\hat{s}$, $\hat{P}$

The TRH relaxes a PPTW by removing temporal constraints on exogenous events (TILs). In the relaxed problem TILs are modelled as instantaneous actions. These may be executed only once at the planner's discretion, and affect states as per the related TIL. The TIL-derived actions are constrained to occur in the order given by the original problem: if TIL e_i occurs earlier in time than e_j in P, then the corresponding TIL-derived actions a_i and a_j are constrained so that a_i is *always* executed before a_j. In detail, each TIL $e_i = \langle t_i, L_i \rangle$ is converted into an instantaneous action, $a_i = \langle Scond(a_i), Eeff(a_i) \rangle$, as follows:

$$\begin{aligned} Scond(a_i) &= p^{a_i} \cup \Theta(e_i) \\ Eeff(a_i) &= \{p_g^{a_i}, \neg p^{a_i}, p^{a_{i+1}}\} \cup L^{a_i} \cup \neg\Theta(e_i) \end{aligned} \quad (1)$$

The auxiliary condition $\Theta(e_i)$ ensures that a_i is executed at most once. $\Theta(e_i)$ is added to the initial state and removed by $Eeff(a_i)$. The effects of the corresponding TIL e_i are given by L^{a_i}. The effect $p_g^{a_i}$ achieves a goal condition for the relaxed problem that ensures the action a_i is executed. This is to conform with PDDL 2.2 semantics, which state all specified TILs must be achieved in a plan.

The auxiliary variables of the form p^{a_i} ensure that execution of TIL-derived actions preserve the TIL order in P. For the earliest TIL, say e_0 corresponding to TIL-derived action a_0, the proposition p^{a_0} is added to the initial state. Breaking ties arbitrarily, each TIL-derived action adds the condition for the next TIL-derived action as part of its effect, $p^{a_{i+1}}$. In this way the temporal ordering of TILs in P is preserved. For example, in Figure 2, this would mean that cargo would still become available for transportation and have a delivery deadline; however the duration of availability is now at the discretion of the planner. Since these windows of availability are not constrained temporally, dead-ends related to scheduling constraints have been removed. The relaxation provides a simple temporal problem that is more amenable to traditional delete relaxed heuristics. In problems made challenging by the time window constraint, determining a solution to the relaxation should be easier. Such a solution, $\hat{\pi}$, is casually valid, but may not be temporally consistent with respect to P, because it ignores the temporal constraints of TILs.

Scheduling the TIL-Relaxed plan $\hat{\pi}$

After finding a plan, $\hat{\pi}$, for the TIL-relaxed problem, the next step is to determine if this plan is also feasible for the original problem, or, if it is not, the magnitude of its violation of the time constraints. This is done by building an STN representation of $\hat{\pi}$, altering temporal constraints to re-introduce the timing of TILs, and checking the STN's consistency. Concretely, we concatenate the relaxed plan with any plan prefix that has already been committed to by the planner to reach the current state, setting the constraints on TIL actions to their original values in the problem, and assessing the temporal consistency of the plan. The STN is built by applying the KK algorithm (Kambhampati and Kedar 1994) to get a deordering of the concatenated plan.

In the relaxed problem, actions describing TILs were allowed to execute at the planners discretion. To determine the consistency of the STN with respect to the original problem, constraints on the execution of these actions must be tightened to reflect the temporal constraint on the original TIL. Figure 3 is the STN of a plan for the TIL-relaxation of our motivating example. Event e_0 denotes the start of plan execution (time zero). Events e_2 and e_{24} represent the start and end, respectively, of the time window during which the cargo must be moved. The TIL constraints position these events exactly with respect to e_0. The precedence constraints from e_2 to e_4 (the start of the loading action) and from e_{23} (the end of a planned unloading action) to e_{24} are causal constraints, formed due to the plan's causal dependency on the TIL's effects.

Consistency of the STN G, with original TIL constraints, is checked by searching for negative cycles in its distance graph, as described by Dechter, Meiri, and Pearl (1991). We use the Shortest Path Faster Algorithm (SPFA) (Fanding 1994) to determine STN consistency.

If G is not consistent, $\hat{\pi}$ is not valid with respect to P and the TRH needs to calculate the degree of violation in order to provide a heuristic evaluation of the state.

Computing the Heuristic Value

When G is not consistent, our aim is to find a minimum relaxation (weakening) of a subset of temporal constraints that make it consistent. The amount of relaxation that is necessary to restore consistency is an indicator of how far the relaxed plan is from being temporally consistent with respect to P. A procedure to compute this minimal relaxation of an inconsistent STN was provided by Yu and Williams (2013); we use their approach. The magnitude of the temporal relaxation is what we use as the heuristic value of the state.

The TRH partitions STN constraints into those that may be relaxed (soft) and those that cannot (hard). Soft constrains are the duration constraints between action start and end events, while all other constraints are hard. The soft constraints are drawn with solid edges in Figure 3.

Yu and Williams (2013) proposed an algorithm to find the optimal relaxation of soft constraints that restore consistency to an STN. This is achieved by iteratively testing for temporal inconsistency, extracting any violated constraints, and posing an LP whose solution gives the magnitude of the necessary relaxation to remove all violations.

In detail, if the STN G is inconsistent, then a conflict will be extracted and represented as a violated inequality constraint. For example, a temporal inequality constraint present in the STN in Figure 3 is that the sum of the lower bounds along the path e_0–e_2–e_4–e_{13}–e_{16}–e_{24} must be less than or equal to the upper bound on e_0–e_{24}. This upper bound is the result of the TIL constraints. The algorithm maintains a set, C, of all discovered inequalities, each of which corresponds to a temporal conflict invalidating $\hat{\pi}$. At each iteration of the loop on lines 7-10, an LP solver finds the minimal weakening of the lower and/or upper bound values required

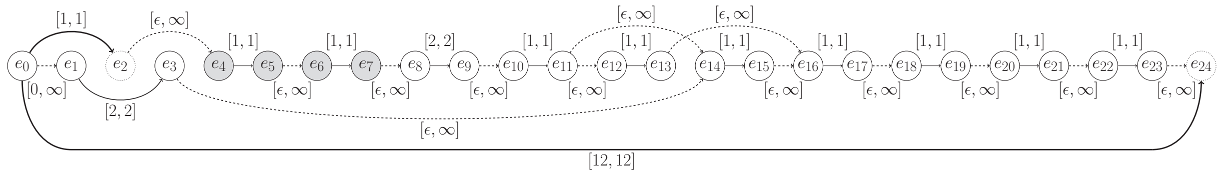

Figure 3: STN for a temporal plan. Events e_2 and e_{24} are TILs, and the constraints $e_2 - e_0 = 1$ and $e_{24} - e_0 = 12$ represent the corresponding time window. Constraints drawn as dashed edges are precedence constraints and those drawn as solid lines are action duration constraints, which may be relaxed when computing the magnitude of violation.

Algorithm 2 TRH Early Termination Subroutine

Input: a temporal network G
Output: $h(s)$, and π, an ordered plan for P

1 **if** G is consistent **then**
2     $\pi \leftarrow \text{EXECUTE}(G)$
3     $h(s) \leftarrow 0.0$
4   **return** $h(s)$, π
5 **end if**
6 **return**

to satisfy all inequalities in C. When G is consistent given the relaxation implied by the solution to the LP, the magnitude of all relaxations in that solution corresponds to the heuristic value. When G remains inconsistent even with the proposed relaxation, C is expanded according to newly discovered conflicts, and iteration continues.

Like Yu and Williams (2013), we use the Incremental Temporal Consistency (ITC) algorithm (Shu, Effinger, and Williams 2005) to detect and extract temporal conflicts. The ITC algorithm has been updated beyond that which was described by Shu, Effinger, and Williams, to use a SPFA for cycle detection and extraction.

From this evaluation of a state, the TRH provides a continuous value that is used by the planner to guide search. Proceeding in the standard way for heuristic search planners, requests for state evaluations will continue for new states encountered until a termination criterion is reached – i.e. a goal is found, all nodes have been encountered, or available computational resources are expended.

Early Termination

During the heuristic evaluation of a state, it could be that the relaxed solution $\hat{\pi}$ is a plan for P. If this does occur, search can be terminated early and a solution returned. This is accomplished with a subroutine that is executed between lines 5 and 6 of Algorithm 1. Algorithm 2 shows this subroutine in detail. Concretely, a consistency check is performed on G (line 1), and if it is consistent then a valid execution of the actions within G is determined (line 2), π. Finally π is returned as a plan for P. We term this extension, TRH with early termination (TRH-ET).

Empirical Evaluation

Implementations We have implemented all variants of the TRH heuristic within the COLIN and POPF planning systems. We refer to these variants of the planners guided by the TRH as COLIN-TRH, COLIN-TRH-ET, POPF-TRH, and POPF-TRH-ET, respectively. COLIN and POPF are temporal planners that support reasoning about TILs. Both use a two stage, state-based search, guided by the Temporal Relaxed Planning Graph (TRPG) heuristic. The first stage uses Enforced Hill Climbing (EHC) search, with helpful action pruning like the FF Planner (Hoffmann and Nebel 2001). If EHC fails to find a solution the planner falls back on a Best-First Search (BFS) in the second stage. All variants of TRH implemented within the COLIN and POPF planning systems use that heuristic in both stages of search. Irrespective of the heuristic used, all systems use helpful action pruning in the EHC stage. To solve the TIL-relaxed problem, $\hat{\pi}$, in the heuristic, we invoke the corresponding original, unmodified planner.[2] All comparisons of the TRH and TRH-ET within the COLIN and POPF planning systems are made against their unmodified equivalents, guided by the TRPG heuristic.

Domains and Problem Sets We evaluate the overall effectiveness of planning systems by the number of problem instances solved within a given time limit (coverage). The effectiveness of heuristics is also measured according to the number of states evaluated and dead end states encountered during search. These observations are made on five temporal domains with time windows. Specifically, the Airport, Pipesworld and Satellite benchmark domains from the 2004 International Planning Competition (IPC), the MMCR domain, and the Crew Planning domain. All available problem instances from the IPC benchmark domains were used. The Crew Planning domain models the problem of deciding how to schedule tasks over a period of days for a crew aboard a space station. It was not originally formulated using TILs, but instead using durative actions modelling the transition from one day to the next. We replaced such actions with TILs encoding the start and end of each day. The Crew Planning variant with TILs is what we use in our evaluation.[3]

We generated 450 problem instances from the MMCR domain with varying number and tightness of the time window constraints. All instances have identical transport networks and numbers of vehicles. Each vehicle can traverse a sub-

[2] We use COLIN to solve $\hat{\pi}$ in all variants of TRH implemented in COLIN, and POPF to solve $\hat{\pi}$ in all variants of TRH implemented in POPF. These planners are themselves guided by the TRPG heuristic.

[3] Both the original and modified problem descriptions are available at: https://github.com/tonyallard/pddl-domains

set of locations. To vary the number of time windows, we generated three sets of scenarios with $n_C \in \{1, 2, 3\}$ cargo delivery requests. Each set contains 30 MMCR scenarios, where cargo origin/destination and vehicle origin have been sampled uniformly from valid locations.

The difficulty of solving PPTWs has been observed to correlate with the tightness of time window constraints (Savelsbergh 1985). To evaluate the impact of time window tightness, each scenario was turned into five instances that differ only by having cargo delivery deadlines of increasing tightness. The duration of each cargo's time window is the minimum action sequence duration required for its delivery, $\mathrm{dur}(A_c)$, multiplied by a factor, τ. (We refer to this factor as "tightness", although it is actually the inverse of tightness.) We used values of $\tau \in \{2, 1.5, 1.25, 1.125, 1.0625\}$, meaning the loosest time windows are twice the minimum delivery time, and the tightest time windows are only 6.25% longer than the duration of an optimal plan found for each cargo item in isolation. This method of setting time windows does not guarantee that the resulting problem is solvable, because the minimum delivery time for each cargo item is calculated ignoring the others. 16 of the MMCR instances were proven unsolvable by at least one of the trialled planners. We remove these problems from the remainder of analysis. Similarly, we also generated three new sets of instances of the Pipesworld domain by scaling down the subgoal deadlines of the original instances in the IPC 2004 set by a factor $\tau \in \{0.9, 0.8, 0.7\}$. Again, we removed instances that were proven unsolvable by at least one of the trialled planners, leaving 25, 22 and 19 instances, respectively.

Experiment Setup Evaluation took place on Dell PowerEdge M620 blade servers, each with 2 Intel Xeon E5-2680 2.70GHz CPUs and 192GB of RAM. Each planner was allowed 30 minutes and 2GB of RAM to solve each problem instance. To account for randomisation in some planner implementations (LPG-TD), we ran each planner on each problem 30 times and report the averages. All plans generated (except LPG-TD, see below) were validated by VAL (Howey, Long, and Fox 2004).

Results Table 1 shows coverage for each domain. Results for the MMCR domain are broken down by tightness and by the number of time windows. The new sets of Pipesworld instances with tightened time windows are likewise shown separately. We also measured the number of states evaluated and number of dead ends encountered during search. Here, a *dead end* is a state from which no relaxed plan can be found. This holds for both the original TRPG and TRH-guided versions of both planners. Table 2 shows the average numbers of states evaluated and dead ends encountered, with the average taken over all instances that were solved by both the baseline and the corresponding TRH-guided planner.

Using the TRH only for search guidance reduces coverage in most domains. The one exception is MMCR, where COLIN-TRH achieves a better result than baseline COLIN on the sets of instances with more or tighter time windows. As Table 2 shows that the amount of search (states evaluated) with TRH is almost uniformly lower, often by one or more orders of magnitude. Therefore this drop in coverage can be attributed to the overhead of invoking the planner on the TIL-relaxed problem for each state evaluation.

The TRH-ET variant results in a significant coverage boost for COLIN in all domains, and equal or better coverage in several domains compared to POPF. The baseline (TRPG-guided) version of POPF is almost always better than baseline COLIN, which is expected since it is built on COLIN and adds several enhancements. Thus, it is noteworthy that COLIN-TRH-ET outperforms both POPF and POPF-TRH-ET in the Pipesworld and MMCR domains, particularly as time windows are tightened. The causes behind the impact of the TRH on the mechanisms that differentiate POPF from COLIN is an area for future investigation. In most domains where one planner version achieves a higher coverage than another the set of solved instances is a superset, but there are a few exceptions: for example, baseline COLIN solves two MMCR instances that COLIN-TRH-ET does not, and POPF-TRH-ET solves one instance of Satellite and two of instances of Pipesworld with $\tau < 1$ that baseline POPF does not.

Results in the MMCR domain have been separated along two dimensions, tightness (τ) and number of time windows (n_C), in order to show the effect of these parameters on planner performance. As time windows tighten, the number of instances solved by all planners decreases, but it decreases more slowly for the planners using TRH, and even more so for the planners with TRH-ET. Recall that MMCR problems were constructed such that instances of the same scenario with different tightness factors differ *only* by the tightness of the time windows, as do the instances of the Pipesworld domain with $\tau < 1$. This allows us to evaluate how performance on the same causal problem drops as time windows are tightened. A factor $\tau = 2$ should provide sufficient slack that the remaining challenge is only solving the underlying causal problem. Looking at coverage at each tightness level as a percentage of the coverage when $\tau = 2$ yields insight into performance degradation due specifically to tightness. Using this ratio, coverage in MMCR at $\tau = \{1.5, 1.25, 1.125, 1.0625\}$ shows that COLIN-TRH-ET degrades at a reduced rate (88.8%, 73.0%, 67.4%, and 66.3%) compared to COLIN-TRH (89.4%, 80.3%, 63.6%, and 62.1%), and COLIN (81.7%, 62.0%, 56.3%, and 54.9%). POPF-TRH-ET shows a similar pattern (86.5%, 73.0%, 67.4%, and 66.3%) over POPF-TRH (79.3%, 72.0%, 53.7%, and 52.4%) and POPF (79.1%, 68.6%, 60.5%, and 59.3%). A slower rate of degradation as time windows tighten can be observed for COLIN also in Pipesworld, using the ratio to the number of original instances solved at each tightness level. Here COLIN-TRH-ET degrades at a reduced rate (44.0%, 31.8%, and 10.5%), compared with COLIN-TRH (12.0%, 9.1%, and 0.0%) and COLIN-TRPG (16.0%, 13.6%, and 5.3%). However, we do not see the same for POPF-TRH-ET. These observations indicate that the TRH-ET improves the robustness of planner performance on PPTWs as time windows increase in number, and as they become tighter.

Table 2 shows that the TRH-guided planners consistently evaluate fewer states and encounter fewer dead ends during search, across commonly solved instances. The TRPG

Table 1: Number of problems solved in each domain. Parenthesis indicate the number of problems in the problem domain. Bold terms indicate if TRH variants exhibit better coverage compared to the based planner with RPG guidance.

Problem Domain	COLIN			POPF			OPTIC		LPG-TD
	TRPG	TRH	TRH-ET	TRPG	TRH	TRH-ET	TRPG	SLFRP	
Airport (50)	9	8	9	8	7	8	8	8	42.10
Crew Planning (30)	14	11	**23**	30	19	30	30	30	11.00
Pipesworld (30)	12	5	**24**	23	12	17	24	0	25.83
τ=0.9 (25)	4	3	**11**	8	2	5	8	0	11.00
τ=0.8 (22)	3	2	**7**	7	1	4	7	0	9.00
τ=0.7 (19)	1	0	**2**	0	0	**1**	3	0	5.00
Satellite (36)	1	1	**3**	5	4	4	10	1	19.33
MMCR τ=2 (90)	71	66	**89**	86	82	**89**	88	0	90.00
τ=1.5 (88)	58	**59**	**79**	68	65	**77**	76	0	84.00
τ=1.25 (86)	44	**53**	**65**	59	59	**65**	64	0	70.00
τ=1.125 (85)	40	**42**	**60**	52	44	**60**	60	0	66.00
τ=1.0625 (85)	39	**41**	**59**	51	43	**59**	58	0	59.00
n_C=1 (150)	149	149	**150**	150	150	150	150	0	150.00
n_C=2 (143)	78	**81**	**124**	106	94	**123**	124	0	128.00
n_C=3 (141)	25	**31**	**78**	60	49	**77**	72	0	91.0

Table 2: Average number of states evaluated (left) and dead ends encountered (right) during search, by domain. Averages are taken over the subset of instances solved by all three configurations of each planner (TRPG, TRH and TRH-ET). Bold entries indicate the best result in each problem domain.

Domain	COLIN						POPF					
	TRPG		TRH		TRH-ET		TRPG		TRH		TRH-ET	
Airport (50)	723.0	201.38	35.0	**0.00**	**1.0**	**0.00**	690.3	282.14	34.1	**0.00**	**1.0**	**0.00**
Crew Plan. (30)	249.6	5.55	28.1	**0.00**	**1.0**	**0.00**	453.1	1.42	36.6	**0.00**	**1.0**	**0.00**
Pipesworld (30)	27947.8	21073.40	2609.6	1178.80	**22.6**	**7.60**	14264.4	8553.83	626.7	260.17	36.9	8.33
τ=0.9 (25)	2811.7	1780.33	1855.7	906.67	36.0	13.33	346.0	150.00	33.0	6.50	**1.0**	**0.00**
τ=0.8 (22)	15863.5	11267.50	29.5	7.00	**1.0**	**0.00**	393.0	181.00	2944.0	1611.00	**1.0**	**0.00**
τ=0.7 (19)	–	–	–	–	–	–	–	–	–	–	–	–
Satellite (36)	195.0	9.00	17.0	**0.00**	**1.0**	**0.00**	617.7	59.67	21.3	**0.00**	**1.0**	**0.00**
MMCR τ=2 (90)	11562.4	4311.77	38.9	6.79	**1.2**	**0.02**	861.2	25.32	53.6	11.54	1.3	0.06
τ=1.5 (88)	13136.1	5061.90	30.6	5.25	**1.0**	**0.00**	745.2	128.05	40.7	8.70	1.4	0.06
τ=1.25 (86)	31869.8	13864.53	31.0	5.74	1.5	**0.09**	220.0	10.45	48.6	13.54	**1.5**	0.11
τ=1.125 (85)	21530.7	10158.39	163.1	36.42	2.8	0.36	216.2	3.41	90.1	28.02	**2.2**	**0.32**
τ=1.0625 (85)	23344.5	10927.26	170.0	38.15	2.9	0.38	205.7	2.65	89.3	28.09	**2.2**	**0.33**
n_C=1 (150)	253.5	57.20	33.7	7.45	**1.3**	**0.08**	238.7	0.53	49.6	12.91	1.4	0.11
n_C=2 (143)	45852.8	18435.12	181.3	37.95	2.8	0.30	246.8	7.58	71.4	22.23	**1.9**	**0.20**
n_C=3 (141)	85495.6	41392.72	61.1	9.06	**1.7**	**0.06**	1877.3	215.29	75.3	15.70	2.0	0.18

heuristic makes limited inference about deadlines, so it often misleads search into causal decisions which ultimately do not yield a schedulable plan, resulting in dead ends. TRH considers deadlines and thus is better able to direct search away from dead ends caused by the time window constraints. This translates into fewer state evaluations. Ending the search when a plan for the TIL-relaxed problem can be scheduled with time windows is a key mechanism. This is how the TRH-ET-guided planners are able to solve a large number of instances with a single call to the heuristic. Regarding the time to find a plan, on commonly solved instances, results are less conclusive. In most domains, the original COLIN and POPF planners are faster; Crew Planning is an exception. Partly, this may be because the commonly solved instances are mainly easy ones. The overhead of compiling the TIL-relaxed problem and invoking a full planner (COLIN and POPF, respectively) involved in each TRH evaluation pays off only on problems that require a longer time to solve. The dramatic reduction in states evaluated and dead ends encountered suggests that greater time savings should be obtainable with a better engineering.

Table 1 includes coverage results for the OPTIC, OPTIC-SLFRP (Tierney et al. 2012), and LPG-TD (Gerevini, Saetti, and Serina 2005) planners.[4] OPTIC is comparable to, sometimes better than, POPF, yet still solves slightly fewer instances of MMCR, at all levels of time window tightness, than both COLIN-TRH-ET and POPF-TRH-ET. The SL-FRP variant behaves like OPTIC in the Airport and Crew Planning domains, but solves almost no problem otherwise. LPG-TD appears to work well in all domains except crew planning, where it fails to parse some instances. A *caveat* here, however, is that plans produced by LPG-TD could not be validated by VAL, because it does not enforce epsilon separation between actions. Thus, for LPG-TD only the coverage is based on the planner's own reported successes, not on the number of plans independently validated.

Related Work

The problem of planning with exogenous events is not new to the planning community. Early work by Vere (1983) investigated modelling such events in DEVISER, a partial order causal link (POCL) planner. DEVISER initialises a partial plan with exogenous events in a pre-processing step. A depth-first search strategy is used to determine actions which achieve goals, and are consistent with the exogenous events. No search guidance was used, and indeed the exploration strategy is described by Vere as a "simulation".

In describing the state-space forward search planner TGP, Smith and Weld (1999) discuss translating exogenous events into dummy actions. A dummy action has no preconditions, and its execution is forced by the planning system whenever considering a time at or beyond the corresponding event time. Several planners, such as CRIKEY3 and COLIN, build on this approach to support problems with time windows. For the purpose of heuristic guidance, dummy actions are indistinguishable from other actions, so such approaches may not foresee scheduling problems caused by time windows as their heuristics do not reason about all temporal constraints.

Constraint-based planners, such as OMPS (Magrelli and Pecora 2010) and ITSAT (Rankooh and Ghassem-Sani 2015), also model exogenous events as actions and rely on a causal plan search engine. They use temporal consistency checking, e.g., via STNs, to detect scheduling conflicts implied by causal planning choices. Such scheduling conflicts may be represented using simple mutex constraints, or rich sequencing constraints expressed compactly as *regular expressions*. Learnt constraints restrict the causal search, to eventually produce a plan that can be scheduled without conflict. However, the weakness of this approach is also that the guidance of the causal plan search does not anticipate conflicts arising from time windows.

Tierney et al. (2012) investigated separating the planning and scheduling components of a PPTW by extracting all temporal constraints and modelling them in a mixed integer program (MIP). They modelled exogenous events as binary variables within the MIP. This technique was implemented in POPF, and its search guidance was still provided by the TRPG heuristic, however with special rules to not relax TILs. Tierney et al. also developed LTOP, an optimal POCL planner capable of reasoning about PPTWs. Similar to POPF the temporal (and optimisation) model was completely abstracted from the action model into a MIP. LTOP is an optimal planner and used domain dependent heuristics to be competitive with a regular MIP model.

[4] We also wanted to include results for the TEMPLM planner, but were unable to obtain the planner or results from its authors.

Marzal, Sebastia, and Onaindia (2014) investigated temporal landmarks as an approach to the related problem of planning with deadlines. Their planner TEMPLM constructs a skeleton plan from temporal landmarks discovered in a step prior to search. During search, candidate plan prefixes are pruned if they violate temporal constraints described by the landmarks. This allows pruning inconsistent causal plans earlier, and thus assists in avoiding dead ends. The pruning information provided by temporal landmarks is orthogonal to the guidance information provided by TRH, and the two could be combined to better tacke hard PPTWs.

The transportation science literature has investigated PPTWs for some time, in the study of Vehicle Routing Problems with time windows (VRPTW) (Solomon 1987). A recent survey of approaches to optimisation in a transportation setting with time windows is given by Hashimoto et al. (2013). In that setting, a number of Large Neighbourhood Searches have been devised which are guided, at least in part, by cost-of-violation style heuristics – i.e. the costs of violating time windows include, but are not limited to: (i) having to pay a driver to sit and wait, (ii) the opportunity cost of missing a delivery, and (iii) bringing on an additional vehicle and driver to ensure all deliveries are made.

Conclusion

We introduced a new planning heuristic, the TRH, for planning problems whose principal difficulty lies in scheduling actions within time windows. Based on solving a relaxed problem without time window constraints, the TRH provides a continuous estimate of the violation of those constraints in the relaxed solution. Implemented in two heuristic-search temporal planners, COLIN and POPF, we showed that the TRH improves planner performance in some domains, in particular as time windows tighten.

The greatest benefit, however, comes from avoiding search altogether when the relaxed solution is a plan also for the problem with time windows. Future work should investigate other uses of the TIL-relaxation and analysis of violations, for example identifying causal constraints that prevent the planner from exploring infeasible areas of the search space.

Acknowledgements This work was supported in part by the ANU–Data61–DST CRP "Automated Planning for Cyber Mission Assurance".

References

Allard, T., and Gretton, C. 2015. A realistic multi-modal cargo routing benchmark. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Coles, A.; Fox, M.; Long, D.; and Smith, A. 2008. Planning with problems requiring temporal coordination. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 892–897.

Coles, A. J.; Coles, A.; Fox, M.; and Long, D. 2010. Forward-chaining partial-order planning. In *ICAPS*, 42–49.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2012. Colin: Planning with continuous linear numeric change. *Journal of Artificial Intelligence Research* 44:1–96.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial intelligence* 49(1):61–95.

Edelkamp, S., and Hoffmann, J. 2004. Pddl2. 2: The language for the classical part of the 4th international planning competition. Technical report.

Fanding, D. 1994. A faster algorithm for shortest-path spfa. *Journal of Southwest Jiaotong University* 2.

Gent, I. P.; MacIntyre, E.; Prosser, P.; Walsh, T.; et al. 1996. The constrainedness of search. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 246–252.

Gerevini, A.; Saetti, A.; and Serina, I. 2005. Integrating planning and temporal reasoning for domains with durations and time windows. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 19, 1226.

Globus, A.; Crawford, J.; Lohn, J.; and Pryor, A. 2004. A comparison of techniques for scheduling earth observing satellites. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 836–843.

Hashimoto, H.; Yagiura, M.; Imahori, S.; and Ibaraki, T. 2013. Recent progress of local search in handling the time window constraints of the vehicle routing problem. *Annals of Operations Research* 204(1):171–187.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (JAIR)*.

Howey, R.; Long, D.; and Fox, M. 2004. Val: Automatic plan validation, continuous effects and mixed initiative planning using pddl. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, 294–301. IEEE.

Kambhampati, S., and Kedar, S. 1994. A unified framework for explanation-based generalization of partially ordered and partially instantiated plans. *Artificial Intelligence* 67(1):29–70.

Kolen, A. W.; Rinnooy Kan, A.; and Trienekens, H. 1987. Vehicle routing with time windows. *Operations Research* 35(2):266–273.

Magrelli, S., and Pecora, F. 2010. A constraint based approach to integrated planning and scheduling. *UK Planning and Scheduling Special Interest Group (PlanSIG)*.

Marzal, E.; Sebastia, L.; and Onaindia, E. 2014. On the use of temporal landmarks for planning with deadlines. In *Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS)*.

Milidiú, R. L.; dos Santos Liporace, F.; and de Lucena, C. J. P. 2003. Pipesworld: Planning pipeline transportation of petroleum derivatives. *Competition Workshop Programme, International Conference on Automated Planning and Scheduling*.

Rankooh, M. F., and Ghassem-Sani, G. 2015. ITSAT: an efficient sat-based temporal planner. *Journal of Artificial Intelligence Research (JAIR)* 53:541–632.

Savelsbergh, M. W. 1985. Local search in routing problems with time windows. *Annals of Operations research* 4(1):285–305.

Shu, I.-h.; Effinger, R. T.; and Williams, B. C. 2005. Enabling fast flexible planning through incremental temporal reasoning with conflict extraction. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 252–261.

Smith, D. E., and Weld, D. S. 1999. Temporal planning with mutual exclusion reasoning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 99, 326–337.

Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research* 35(2):254–265.

Tierney, K.; Coles, A.; Coles, A.; Kroer, C.; Britt, A. M.; and Jensen, R. M. 2012. Automated planning for liner shipping fleet repositioning. In *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press.

Vere, S. A. 1983. Planning in time: Windows and durations for activities and goals. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (3):246–267.

Yu, P., and Williams, B. 2013. Continuously relaxing overconstrained conditional temporal problems through generalized conflict learning and resolution. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2429–2436. AAAI Press.