Challenges in Maintaining Minimal, Decomposable Disjunctive Temporal Problems

James C. Boerkoel Jr. and Edmund H. Durfee
Computer Science and Engineering, University of Michigan
Ann Arbor, MI
{boerkoel,durfee}@umich.edu

Abstract

In many scheduling applications, new scheduling constraints can arise dynamically due to exogenously determined events and preferences. In such environments, maintaining a set of all possible remaining schedules can be much more robust than selecting a single schedule. In this paper we discuss two properties, minimality and decomposability, that are necessary for faithfully representing the set of all remaining solutions to a Disjunctive Temporal Problem (DTP). We prove minimal and decomposable representations always exist for a consistent DTP. We also introduce metrics for comparing different minimal, decomposable DTP representations in terms of space and time efficiency and propose ideas for improving efficiency based on work from dispatching disjunctive schedules.

1 Introduction

In many scheduling applications, the actual duration of a specific activity, such as the transportation time between two locations, may be either uncertain, exogenously determined, or subject to unexpressed preferences. As an example, consider a truck that starts out at a depot and must make deliveries to three different locations by a predetermined deadline. While the truck can visit the three locations in any order, each order may have different implications on travel and processing time due to traffic congestion and the overhead involved in reshuffling inventory on the truck. In such applications, calculating a single plan with a specific schedule may be brittle to the dynamics and preferences involved in the problem. A more robust approach for dealing with uncertainty, dynamism, and unexpressed preferences in logistics and scheduling applications is to instead calculate the set of *all* feasible schedules. This approach efficiently supports queries of the form "When can I perform delivery A?", or "Can I make delivery X before I make delivery Y?".

However, the set of all feasible schedules generally grows exponentially in size as the number of events increases. Fortunately, there exist constraint-based problem formulations that are capable of *compactly* representing sets of feasible schedules [Dechter *et al.*, 1991; Stergiou and Koubarakis, 2000; Shah and Williams, 2008]. The most general of these is called the Disjunctive Temporal Problem (DTP), whose generality is required for representing the example problem described above. Faithfully representing the set of *all feasible* solutions requires establishing properties called *minimality* and *decomposability*. Minimality ensures that the complete set of solutions is represented while decomposability ensures that any consistent, partial schedule that respects constraints can be soundly extended to a complete solution schedule.

In this paper, we will review three important constraint-based scheduling problem formulations, including the Disjunctive Temporal Problem (DTP) in Section 2. We will demonstrate that the concepts of minimality and decomposability, which are well defined for some problem formulations, extend naturally to the more general DTP. We will also prove in Section 3 that despite fundamental differences from their less-complex predecessors, minimal, decomposable representations for consistent DTPs always exist. In Section 4 we explore the challenges associated with establishing minimal, decomposable DTP representations that are more efficient in terms of space (compactness) and the speed of reestablishing minimality and decomposability after an update. Gaining inspiration from important related work in dispatching disjunctive scheduling, we will identify specific challenges to, and propose ideas for, more efficient maintenance. Finally, we will conclude with some discussion and future work in Section 5.

2 Background

In this section we provide definitions necessary for understanding our contributions.

2.1 Simple Temporal Problem

We begin by adapting our description of the Simple Temporal Problem (STP) [Boerkoel and Durfee, 2011]. As defined by Dechter *et al.* [1991], the Simple Temporal Problem (STP), $\mathcal{S} = \langle V, C_{STP} \rangle$, consists of a set of timepoint variables, V, and a set of temporal difference constraints, C_{STP}. Each timepoint variable represents an event, and has an implicit, continuous numeric domain. Each temporal difference constraint c_{ij} is of the form $v_j - v_i \leq b_{ij}$, where v_i and v_j are distinct timepoints, and $b_{ij} \in \mathbb{R}$ is a real number bound on the difference between v_j and v_i.

To exploit extant graphical algorithms and efficiently reason over the STP *constraint network*, each STP is associated with a weighted, directed graph, $\mathcal{G} = \langle V, E \rangle$, called a *distance graph*.

The set of vertices V is as defined before (each timepoint variable acts as a vertex in the distance graph) and E is a set of directed edges, where, for each constraint c_{ij} of the form $v_j - v_i \leq b_{ij}$, we construct a directed edge, e_{ij} from v_i to v_j with an initial weight $w_{ij} = b_{ij}$. As a graphical short-hand, each edge from v_i to v_j is assumed to be bi-directional, compactly capturing both edge weights with a single label, $[-w_{ji}, w_{ij}]$, where $v_j - v_i \in [-w_{ji}, w_{ij}]$ and a weight w_{xy} is initialized to ∞ if there exists no corresponding constraint, $c_{xy} \in C_{STP}$. All times (e.g. 'clock' times) can be expressed relative to a special ***zero*** timepoint variable, $z \in V$, that represents the "start of time". Bounds on the difference between v_i and z can be thought of as "unary" constraints specified over a timepoint variable v_i. Moreover, w_{zi} and w_{iz} then represent the earliest and latest times, respectively, that can be assigned to v_i, and thus implicitly define v_i's domain. In this paper, we will assume that z is always included in V and that, during the construction of $\mathcal{G}$, an edge e_{zi} is added from z to every other timepoint variable $v_i \in V$. Examples of distance graphs that correspond to solutions for the example problem introduced in Section 1 (and formalized in Section 2.2) are in Figure 1. An STP is *consistent* if it contains at least one *solution*, which is an assignment of specific time values to timepoint variables that respects all constraints to form a *schedule*. Approaches for efficiently finding and representing the set of *all* solutions is presented in Section 2.4.

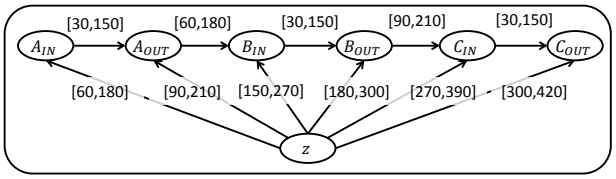
2.2 Disjunctive Temporal Problem

In order to represent a scheduling problem as an STP, the relative, partial order of all involved timepoints must be predetermined. However, many scheduling problems require making decisions over the relative order in which to execute activities. For example, capturing the fact that the truck can visit locations in *any* order requires *disjunctive* constraints. In comparison to the STP, the Disjunctive Temporal Problem (DTP), $\mathcal{D} = \langle V, C_{DTP} \rangle$, specifies a more general set of disjunctive constraints, C_{DTP}, where $c \in C_{DTP}$ represents a disjunction over a set of *any* temporal difference constraints. A constraint c takes the form $d^1 \vee d^2 \vee \cdots \vee d^k$ for some $k \geq 1$, where each disjunct d represents a typical temporal difference constraint over (possibly *different*) pairs of timepoints, $v_j - v_i \leq b_{ij}$. Note that an STP is a special case of a DTP where $k = 1$ for all constraints ($C_{STP} \subseteq C_{DTP}$ and subsequently, STP $\subseteq$ DTP).

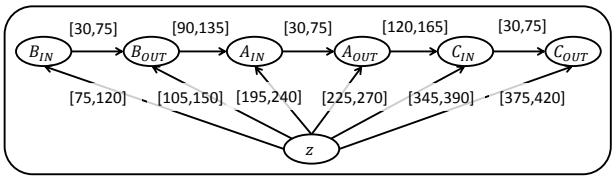
Equipped with the more general DTP representation, we now formally illustrate how to faithfully capture the constraints and other aspects of a more detailed version of the example logistics problem introduced in Section 1. We summarize this representation in Table 1. The problem involves a single truck that needs to make deliveries to three locations, A, B, and C. In addition to the zero timepoint z (where $z = 0$ represents the start time of the journey), there are timepoint variables representing the arrival X_{IN} and departure X_{OUT} of the truck to location X, for each of the three locations. The truck starts its day at a relatively centrally-located depot, and must make each delivery by various deadlines ($X_{OUT} - z \leq b_{DL(X)}$ $\forall X$ where $b_{DL(X)}$ is delivery X's deadline). Each location requires at least 30 minutes duration for unloading ($X_{IN} - X_{OUT} \leq -30$ $\forall X$). Note that constraints over transition

Trans. from Depot	Deadline	Min. Duration
$z - A_{IN} \leq -60;$	$A_{OUT} - z \leq 300;$	$A_{IN} - A_{OUT} \leq -30;$
$z - B_{IN} \leq -75;$	$B_{OUT} - z \leq 360;$	$B_{IN} - B_{OUT} \leq -30;$
$z - C_{IN} \leq -90;$	$C_{OUT} - z \leq 420;$	$C_{IN} - C_{OUT} \leq -30;$
Location to Location Transition (disjunctive)		
$A_{OUT} - B_{IN} \leq -60 \vee B_{OUT} - A_{IN} \leq -90;$ $B_{OUT} - C_{IN} \leq -90 \vee C_{OUT} - B_{IN} \leq -120;$ $A_{OUT} - C_{IN} \leq -120 \vee C_{OUT} - A_{IN} \leq -150$		

Table 1: Summary of the example logistics problem.



a) The minimal, decomposable STP distance graph corresponding to labeling: $\ell = \{A_{OUT} - B_{IN} \leq -60; B_{OUT} - C_{IN} \leq -90; A_{OUT} - C_{IN} \leq -120\}$



b) The minimal, decomposable STP distance graph corresponding to labeling: $\ell = \{B_{OUT} - A_{IN} \leq -90; B_{OUT} - C_{IN} \leq -90; A_{OUT} - C_{IN} \leq -120\}$

Figure 1: The distance graphs corresponding to minimal, decomposable solutions to the example problem (Table 1) corresponding to the three feasible labelings.

time (e.g., constraints of the form $X_{OUT} - Y_{IN} \leq b_{YX} \vee Y_{OUT} - X_{IN} \leq b_{XY}$ for some locations X and Y and bounds b_{YX} and b_{XY}), which include transportation time, are neither reflexive nor transitive due to traffic congestion and overhead involved in reshuffling inventory on the truck.

The DTP is often solved using a *meta-CSP* formulation, where each disjunctive temporal constraint $c \in C_{DTP}$ forms a *meta-variable* with a domain of *meta-values* composed of the set of possible disjuncts. Following the notation of Dechter *et al.* [1991], a *labeling*, ℓ, of a DTP, $\mathcal{D}$, is an STP formed by selecting one meta-value (disjunct) for each meta-variable (disjunctive constraint). A schedule s, then, is a solution to $\mathcal{D}$ if and only if it is a solution to one of $\mathcal{D}$'s labelings. For general DTPs, there are $O(k^{|C_{DTP}|})$ possible labelings, each of which must be explored in the worst case, making the DTP an NP-hard problem [Stergiou and Koubarakis, 2000]. In this particular example problem, of the $2^3 = 8$ possible labelings, only two ($A \rightarrow B \rightarrow C$ and $B \rightarrow A \rightarrow C$; which correspond to the STPs in Figure 1 (a) and (b) respectively) satisfy all scheduling constraints.

A *singleton* constraint, $c \in C^{k=1}_{DTP}$, is one that contains only a single disjunct. Tsamardinos and Pollack [2003] note that the subset $C^{k=1}_{DTP} \subseteq C_{DTP}$ can be used to form an STP $\langle V, C^{k=1}_{DTP} \rangle$. This STP can be used to compile new and tighter singleton constraints that constrain which meta-values can be assigned to which meta-variables. This forward-checking procedure prunes any disjunct that is inconsistent with the STP compilation, since it is guaranteed to be inconsistent with

the overall DTP. This pruning process may result in more constraints being added to the set $C^{k=1}_{DTP}$, which further tightens the STP compilation, possibly leading to more pruning. In the extreme, this process could prune until all disjunctive temporal constraints are singleton, eliminating the need for combinatorial search in the meta-CSP.

More generally, the meta-CSP formulation leads to a search algorithm that interleaves the STP forward-checking procedure with an assignment of a meta-value to a meta-variable. This has the effect of growing the set $C^{k=1}_{DTP}$ and incrementally tightening the corresponding STP compilation. If a particular assignment of a meta-value d_i to a meta-variable c_i leads to an inconsistent STP instance, that assignment is backtracked. Since at this point d_i is known to be inconsistent with the current STP compilation, a procedure known as *semantic branching* allows the STP relaxation to be tightened by adding d_i's inverse implication. Expressing these otherwise implicit constraints further tightens the STP relaxation, which in turn can lead to improved forward checking performance. Additionally, Tsamardinos and Pollack [2003] describe how to incorporate CSP techniques such as no-good recording and backjumping into the meta-CSP search algorithm to further decrease DTP solution algorithm runtime.

2.3 Temporal Constraint Satisfaction Problem

A Temporal Constraint Satisfaction Problem(TCSP), $\mathcal{T} = \langle V, C_{TCSP} \rangle$, is a special case of a DTP ($C_{STP} \subseteq C_{TCSP} \subseteq C_{DTP}$ and thus $STP \subseteq TCSP \subseteq DTP$) where each constraint $c \in C_{TCSP}$ takes the form $v_j - v_i \in [-b^1_{ji}, b^1_{ij}] \vee \ldots \vee [-b^k_{ji}, b^k_{ij}]$. That is, all disjuncts specify bounds over the difference between the *same* two timepoints. Similarly to the DTP, each of $O(k^{|C_{TCSP}|})$ possible labelings may need to be explored before finding a solution, making the TCSP an NP-hard problem [Dechter *et al.*, 1991]. While there exist procedures for transforming a DTP to a TCSP [Planken, 2007], the TCSP is more limited in the problems it can *naturally* represent. For example, the disjunctive constraints from the simple running example problem involve *different* pairs of variables (Table 1, lower), which the TCSP is not directly able to represent.

2.4 Minimality and Decomposability

Dechter *et al.* [1991] define both minimality and decomposability in terms of temporal constraint networks such as the STP and the TCSP. These definitions extend quite naturally to the DTP. A ***minimal*** constraint c_{ij} is one whose interval(s) correctly specify the set of *all* feasible values for the difference $v_j - v_i$. Similarly, a variable with a minimal domain is one whose constraints with the zero timepoint are minimal. A DTP is minimal if and only if all of its constraints and timepoint domains are minimal. A DTP is ***decomposable*** if any locally consistent assignment of a set of timepoint variables can be extended to a solution. Each of the STPs presented in Figure 1 are both minimal and decomposable.

In short, a minimal and decomposable DTP faithfully represents the complete and sound set of solutions by establishing the tightest bounds on timepoint variables such that: (1) no solutions are eliminated and (2) any assignment of a specific time to a timepoint variable (or bound to a constraint) that respects these bounds can be extended to a solution using a backtrack-free search. Establishing minimality and decomposability allows efficient processing of queries such as "at what time can activity X be performed" and "what are the potential relationships between activity X and Y". Unfortunately, establishing minimality and decomposability for general, disjunctive scheduling problems, such as the TCSP and DTP, is NP-hard [Dechter *et al.*, 1991].

The STP presents a special case where minimality and decomposability can be established efficiently (in $O(|V|^3)$) by applying an all-pairs-shortest-path algorithm, such as Floyd-Warshall [1962], to the distance graph to find the tightest possible path between every pair of timepoints, v_i and v_j, forming a fully-connected graph, where $\forall i, j, k, w_{ij} \leq w_{ik} + w_{kj}$. The resulting graph is then checked for consistency by validating that there are no negative cycles, that is, $\forall i \neq j$, ensuring $w_{ij} + w_{ji} \geq 0$ [Dechter *et al.*, 1991]. Recent work exploits sparsity in the constraint network to establish minimality and decomposability more efficiently [Xu and Choueiry, 2003; Shah and Williams, 2007; Planken *et al.*, 2008].

Since establishing minimality and decomposability in disjunctive temporal problems is NP-hard, much work has focused on efficient, polynomial-time methods to increase the level of consistency [Dechter *et al.*, 1991; Stergiou and Koubarakis, 2000; Tsamardinos and Pollack, 2003; Choueiry and Xu, 2004]. These consistency improvements are a partial step towards establishing minimal and decomposable representations and can be used as a preprocessing or constraint propagation technique during a meta-CSP search to find a component STP solution. Next we will prove that minimal, decomposable representations for consistent DTPs always exist.

3 The Existence of Minimal, Decomposable DTP Representations

In this section, we prove that the definitions of minimality and decomposability do indeed extend to DTPs by proving that a minimal and decomposable representation for a consistent DTP always exists. Despite the similarities between the TCSP and the DTP, there are challenges to extending the concepts of minimality and decomposability to the DTP. The heart of these challenges stem from the fact that DTP constraints can be specified over arbitrarily many different pairs of timepoint variables. While in a TCSP, it is well-defined whether or not there is a constraint that must necessarily be enforced between a pair of timepoint variables, this is not the case in DTPs. From the running example problem, the temporal difference constraint between A_{IN} and B_{OUT} only needs to be enforced if the temporal difference constraint between B_{IN} and A_{OUT} is not enforced, and *vice-versa*. That is, the structure of a minimal, decomposable temporal constraint network for a TCSP is obvious *a priori* (since disjunctive choices are always still between the *same* pair of timepoints), while the structure of a minimal, decomposable temporal constraint network for the more general DTP is not.

Notice that the set of solutions to both the DTP and TCSP can be represented as a set of minimal, decomposable STPs. To generate this set, we can naïvely enumerate each of the DTP's (exponentially many) feasible labelings, ℓ and then

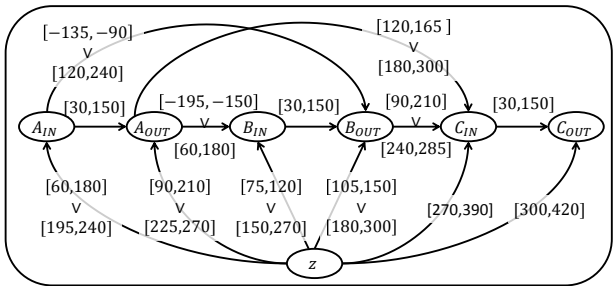


Figure 2: A TCSP representing the minimal network associated with the problem in Table 1.

calculate the minimal, decomposable STP associated with each ℓ. We exploit this observation in our proof that minimal representations of DTPs always exist, which follows as a series of corollaries to Dechter *et al.*'s proof (omitted for brevity) of Theorem 1:

Theorem 1. *[Dechter* et al.*, 1991] The minimal network, $\mathcal{M}$, of a given TCSP, $\mathcal{T}$, satisfies $\mathcal{M} = \cup_\ell M_\ell$, where M_ℓ is the minimal network of the STP defined by labeling ℓ, and the union is over all possible labelings.*

Corollary 1. *The minimal network, $\mathcal{M}$, of a given DTP, $\mathcal{D}$, satisfies $\mathcal{M} = \cup_\ell M_\ell$, where M_ℓ is the minimal network of the STP defined by labeling ℓ, and the union is over all possible labelings.*

Proof. Follows *mutatis mutandis* from Theorem 1. □

Corollary 2. *A minimal representation of a consistent DTP always exists.*

Proof. The minimal network, $\mathcal{M}$, of a DTP, $\mathcal{D} = \langle V, C \rangle$, can always be formulated as the TCSP, $\mathcal{T} = \langle V, C_\mathcal{M} \rangle$, where the set of constraints, $C_\mathcal{M}$, is composed of constraints $c_{ij} \in C_\mathcal{M}$ defined as $v_j - v_i \in \cup_\ell (\mathcal{M}_\ell)_{ij}$, where $(\mathcal{M}_\ell)_{ij}$ corresponds to the bound interval on the difference between v_j and v_i in the minimal network of the STP corresponding to label ℓ. □

Figure 2 represents the TCSP that forms the minimal network for the example problem presented in Table 1. Notice that we include every edge that shows up in either of the solution STPs. For example, the edge between A_{IN} and B_{OUT} is included with both a negative and positive interval, capturing the cases where A occurs before B and *vice-versa*, respectively. Note that in general an algorithm that assigns timepoints using the minimal network alone does is *not* guaranteed decomposability. For example, if we assign the duration of any of the activities represented in Figure 2 to 80, our future assignments should be limited according to selecting the STP in Figure 1 (a); however, the propagation of constraints in the minimal network alone does not guarantee that this would occur.

While the original definition of the decomposability property — a temporal network where any locally consistent assignment of a set of timepoint variables can be extended to a solution — extends naturally to the DTP, it is immediately not obvious whether or not decomposability can always be established for a DTP for a couple of reasons. First is that a DTP involves constraints with high cardinality, which, in general, can be much harder to decompose than a problem exclusively containing binary constraints [Gent *et al.*, 2000]. A second, related reason is that how the set of timepoint variables is assigned influences whether or not certain temporal difference constraints need be enforced in the set's complement. However, we again exploit our naïve representation to prove:

Theorem 2. *A decomposable representation of a consistent DTP always exists.*

Proof. If a DTP is consistent, its set of solutions can be represented as a set of minimal, decomposable STPs corresponding to each feasible labeling, ℓ. Given this representation, any assignment to a set of variables locally consistent with at least one of these STPs, by definition of a decomposable STP, is guaranteed to be extensible to a solution. □

This theorem leads naturally to a procedure for assigning variables in a locally consistent way. First, a variable can only be assigned a value if it appears in its domain (or alternatively, a constraint can only be assigned a bound within one of its intervals of possible bounds) in at least one minimal, decomposable solution STP. Second, once an assignment is made, all STPs that are inconsistent with this assignment are pruned, thereby guaranteeing that subsequent assignments will be locally consistent within one or more minimal, decomposable solution STPs.

In this section, we demonstrated that minimal, decomposable representations always exist for consistent DTPs. However, in general, enumerating every minimal, decomposable STP for each of the exponentially-many consistent labelings is neither compact nor efficient to reason over. In the next section we explore the challenges for more efficiently representing and establishing minimal, decomposable DTPs.

4 An Efficient Minimal, Decomposable DTP Representation

While we have shown that minimal, decomposable representations for consistent DTPs always exist, all representations are not necessarily created equally. As discussed earlier, one of the main advantages of a minimal, decomposable DTP representation is to support *queries* that ask "when can I perform activity X" either in general, or relative to another timepoint. However, presumably if one is interested in posing such queries, one is also interested in both making informed scheduling decisions, and perhaps performing additional queries in the future. So in this sense, we are also interested in how efficiently a minimal, decomposable DTP can be *updated*. Finally, given the exponential nature of the problem, in many scenarios, *space* requirements of the representation may also be of concern. In summary, we can compare representations in terms of (1) *query efficiency*, (2) *update efficiency*, and (3) *space efficiency*.

While our construction of the minimal network related to a DTP may support efficient queries, overall, using a possibly exponential number of STPs to represent and maintain (e.g. update) a decomposable DTP is likely to be quite inefficient in general. These issues lead to a natural question: are there

better ways for establishing and representing minimal, decomposable DTPs than the naïve approaches described in Section 3? We turn to some important related work in dispatching disjunctive schedules for insights into answering this. While not all goals of dispatchable execution align perfectly to those of maintaining minimal, decomposable DTPs, a dispatch agent must make fast recommendations (query efficiency, update efficiency) and may also have limited space capabilities (space efficiency).

4.1 Related Work: Fast Dispatch of Disjunctive Schedules

A minimal, decomposable STP instance naturally lends itself to *dispatchable execution* — an online approach whereby a dispatcher efficiently adapts to scheduling upheavals by imposing additional, restrictive constraints by scheduling timepoints immediately prior to execution [Muscettola *et al.*, 1998]. Shah and Williams [2008] generalize this idea to disjunctive scheduling problems by calculating a *dispatchable* representation of a TCSP. While a previous approach for dispatching DTPs exists [Tsamardinos *et al.*, 2001], this approach is similar in spirit to the naïve approach described in Section 3 by calculating and maintaining the set of all solution STPs. However, Shah and Williams [2008] recognize that many of the solution schedules contain significant redundancy, and so compactly represent the set of decomposable STP solution instances in terms of just their differences. This approach leads to not only a much more compact representation, but also faster execution by avoiding the need to simultaneously and separately update each disparate STP. The algorithm propagates each disjunct using a recursive, incremental constraint compilation technique, and for each disjunct, a list of logical conclusions is kept. Additionally, a list of conflicts is maintained as inconsistencies arise. Their basic execution dispatch algorithm adds each timepoint without any predecessors to an event list, and then as one of these timepoints becomes 'live' (when the current time falls within timepoint's domain), it is selected and updated to occur at the current time, after which the update is propagated throughout the remaining problem. They demonstrate empirically that their approach leads not only to orders of magnitude more compact representations, but also to orders of magnitude faster execution than the suggested naïve approach.

4.2 Extending to the DTP

Since the set of solutions for both DTPs and TCSPs can be represented by a set of minimal, decomposable STP solutions, Shah and Williams' approach for compactly representing the set of solutions by eliminating redundant information is naturally extensible to the DTP. In fact, note that the compact list of implied relationships (of the form $v_j - v_i \in [-b_{ji}, b_{ij}] \rightarrow v_y - v_x \in [-b_{yx}, b_{xy}]$) and conflicts (of the form $\neg v_j - v_i \in [-b_{ji}, b_{ij}] \vee \ldots \vee \neg v_y - v_x \in [-b_{yx}, b_{xy}]$) output by Shah and Williams' approach requires the generality of a DTP constraint to represent (since they are constraints involving *different* pairs of variables). This leads to a more general observation:

Observation 1. *Efficiently representing a set of minimal, decomposable STPs in conjunctive normal form (CNF) requires the representational power of a DTP.*

Constraints for Figure 1 (a)	**Constraints for Figure 1 (b)**
$A_{OUT} - B_{IN} \leq -60 \rightarrow$ $(B_{OUT} - A_{IN} \leq -90 \vee)$	$B_{OUT} - A_{IN} \leq -90 \rightarrow$ $(A_{OUT} - B_{IN} \leq -60 \vee)$
$A_{IN} - z \leq 180;$	$z - A_{IN} \leq -195;$
$A_{OUT} - z \leq 210$	$z - A_{OUT} \leq -225;$
$z - B_{IN} \leq -150$	$B_{IN} - z \leq 120;$
$z - B_{OUT} \leq -180$	$B_{OUT} - z \leq 150;$
$A_{OUT} - C_{IN} \leq 165;$	$C_{IN} - A_{OUT} \leq -180;$
$C_{IN} - B_{OUT} \leq -240;$	$B_{OUT} - C_{IN} \leq 210;$
	$A_{OUT} - A_{IN} \leq 75;$
	$B_{OUT} - B_{IN} \leq 75;$
	$C_{OUT} - C_{IN} \leq 75;$
	$z - C_{IN} \leq -345;$
	$z - C_{OUT} \leq -375;$

Table 2: A minimal, decomposable representation of the example logistics problem.

In addition to the constraints displayed in Figure 2, the constraints presented in Table 2 guarantee both minimality and decomposability for the example logistics problem. Following the same principle as Shah and Williams, we capture these implied relationships compactly by noting that the only difference in the labelings between the Figure 1 (a) and Figure 1 (b) are the labels $A_{OUT} - B_{IN} \leq -60$ (which implies the STP encoded by the temporal difference constraints in the first column of Table 2) and $B_{OUT} - A_{IN} \leq -90$ (which implies the STP encoded by the temporal difference constraints in the second column of Table 2). As a result, any update that is not common to both solution STPs will result in the the correct labeling being applied during forward-checking, which in turn results in a decomposable, minimal network after propagating the constraints. Whereas our representation required a linear (in the number of STP solution edges) number of additional constraints, to generally represent the two solutions encoded in Figure 1 in CNF form would require representing a combinatorial number of additional disjunctive constraints.

4.3 Open Challenges

To this point, we have exploited the DTP's similarity to the TCSP to address many important challenges associated with maintaining minimal, decomposable DTP representations. Particularly, we have shown that we can conceptually extend Shah and Williams' framework to calculate minimal, decomposable DTP representations, but minimality and decomposability only guarantee query efficiency. We now identify important differences between the DTP and TCSP that lead to unique challenges in efficiently establishing and maintaining minimality and decomposability in DTPs.

At a high level, Shah and Williams' approach is roughly similar to the meta-CSP search described in Section 2.2, however there are many important key differences that pose unique challenges for update efficiency. First, when propagating constraints in a TCSP, forward-checking only directly affects the constraint that is currently being propagated. However, in a DTP, pruning a meta-value during forward-checking could lead to a unary meta-variable (disjunctive constraint with only one remaining feasible temporal difference constraint) [Stergiou and Koubarakis, 2000]. This in turn could lead to a new constraint being posted in a distant, non-neighboring portion of the temporal network. That is, propagation jumps around the temporal network rather than only following links through

the network. A second, related difference is that DTP search decisions themselves also can fundamentally change the structure of the temporal constraint network by deciding to add temporal difference constraint between one pair of timepoints instead of some other, different pair. Both of these differences pose challenges to update efficiency, since it is no longer possible to systematically propagate constraints using the temporal network alone. Instead, updates must be simultaneously propagated through both the low-level, minimal temporal network and also the high-level, meta-level CSP. Incorporating support for general, decomposability in the meta-CSP requires fundamental changes to Shah and Williams' algorithm to efficiently learn and maintain all implied relationships and no-good constraints at the meta-level.

Together, these two differences pose a third challenge: can we encode the differences between two solution STPs as compactly, when the differences between them are more systematic in nature? Recall that Shah and Williams' basic approach attempts to exploit redundant information. However, when search decisions and constraint propagation lead to temporal networks with significant differences in structure (e.g., STPs that contain different constraints between different pairs of timepoints), it is unclear how compact, in general, the representation that this approach generates will be.

Finally, up to this point, when we discuss update efficiency, we have largely meant returning a perturbed minimal, decomposable DTP back to a minimal and decomposable state. However, in many applications, the time it takes to establish the initial minimal, decomposable representation may be critical. Shah and Williams' approach is intended as a precompilation approach, and so is never evaluated in terms of initial compilation time. We conjecture that incorporating recent CSP-based search techniques, such as forward-checking, no-good learning, etc. [Tsamardinos and Pollack, 2003], and also SAT-based techniques, such as unit-propagation and a two-literal watching, etc. [Armando *et al.*, 2004; Nelson and Kumar, 2008] could speed the overall process of establishing a decomposable by orders of magnitude, but also poses significant algorithmic engineering challenges.

5 Discussion

In this paper, we demonstrated how the concepts of minimality and decomposability, which had previously been formally defined for only the STP and TCSP, can naturally be extended to the more general DTP formulation to represent the complete, sound set of solutions. We contributed proofs that minimal and decomposable representations of consistent DTP always exist, though not necessarily in an efficient, compact form. We introduced metrics for comparing different minimal, decomposable DTP representations in terms of efficiency, including compactness (space) and query and update speed. We then both discussed the challenges of and offered insights for extending an approach for incrementally compiling TCSPs for fast schedule dispatching to the more general DTP. An interesting extension of this work would be an algorithm that can compute and maintain minimal and decomposable DTP representations in a more incremental or anytime manner for applications that cannot afford to wait for costly precompilation algorithms to complete. Our vision is that the insights of this work can lead to online algorithms that effectively and efficiently adapt to scheduling eventualities that arise in real time and do so with minimal space requirements. These algorithms would have significant implications for logistics applications where the pace of scheduling perturbations may outstrip a scheduler's ability to replan or reschedule, while also granting more autonomy for practitioners to, on-the-fly, select the schedules and plans that best suit their immediate needs and preferences.

References

[Armando *et al.*, 2004] A. Armando, C. Castellini, E. Giunchiglia, and M. Maratea. A SAT-based Decision Procedure for the Boolean Combination of Difference Constraints. In *Proc. of SAT'04*, pages 166–173, 2004.

[Boerkoel and Durfee, 2011] J.C. Boerkoel and E.H. Durfee. Distributed Algorithms for Solving the Multiagent Temporal Decoupling Problem. In *Proc. of AAMAS 2011*, pages 141–148, 2011.

[Choueiry and Xu, 2004] B.Y. Choueiry and L. Xu. An efficient consistency algorithm for the temporal constraint satisfaction problem. *AI Communications*, 17(4):213–221, 2004.

[Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. In *Knowledge representation*, volume 49, pages 61–95. The MIT Press, 1991.

[Floyd, 1962] R.W. Floyd. Shortest path. *Communications of the ACM*, 5(6):345, 1962.

[Gent *et al.*, 2000] I. Gent, K. Stergiou, and T. Walsh. Decomposable Constraints. *Artificial Intelligence*, 123(1-2):133–156, 2000.

[Muscettola *et al.*, 1998] N. Muscettola, P. Morris, and I. Tsamardinos. Reformulating temporal plans for efficient execution. In *Proc. of KR'98*, pages 444–452, 1998.

[Nelson and Kumar, 2008] Blaine Nelson and T. K. Satish Kumar. CircuitTSAT: A solver for large instances of the disjunctive temporal problem. In *Proc. of ICAPS-08*, pages 232–239, 2008.

[Planken *et al.*, 2008] L. Planken, M. de Weerdt, and R. van der Krogt. P3C: A new algorithm for the simple temporal problem. In *Proc. of ICAPS-08*, pages 256–263, 2008.

[Planken, 2007] Leon R. Planken. Temporal reasoning problems and algorithms for solving them (literature survey). Literature survey, Delft University of Technology, October 2007.

[Shah and Williams, 2007] J.A. Shah and B.C. Williams. A Fast Incremental Algorithm for Maintaining Dispatchability of Partially Controllable Plans. In *Proc. of ICAPS-07*, 2007.

[Shah and Williams, 2008] J.A. Shah and B.C. Williams. Fast Dynamic Scheduling of Disjunctive Temporal Constraint Networks through Incremental Compilation. In *Proc. of ICAPS-08*, 2008.

[Stergiou and Koubarakis, 2000] K. Stergiou and M. Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, 120(1):81–117, 2000.

[Tsamardinos and Pollack, 2003] I. Tsamardinos and M.E. Pollack. Efficient solution techniques for disjunctive temporal reasoning problems. *Artificial Intelligence*, 151(1-2):43–89, 2003.

[Tsamardinos *et al.*, 2001] I. Tsamardinos, M.E. Pollack, and Ganchev P. Flexible Dispatch of Disjunctive Plans. In *Proc. of ECP-06*, pages 417–422, 2001.

[Xu and Choueiry, 2003] L. Xu and B Choueiry. A new effcient algorithm for solving the simple temporal problem. In *Proc. of TIME-ICTL-03*, pages 210–220, 2003.