CHAPTER 2

The Multiagent Simple Temporal Problem

2.1 Introduction

The Simple Temporal Problem (STP) formulation is capable of representing scheduling problems, along with their corresponding solution spaces, where if the order between pairs of activities matters, this order has been predetermined. As such, the STP acts as the core scheduling problem representation for (and flexible representation of scheduling solutions to) many interesting planning problems (Laborie & Ghallab, 1995; Bresina et al., 2005; Castillo et al., 2006; Smith et al., 2007; Barbulescu et al., 2010). Likewise, the Multiagent STP (MaSTP) is as a multiagent, distributed representation of scheduling problems and their solutions for multiagent and can be used for multiagent plan execution and monitoring.

As an example of this type of problem, suppose Ann, her friend Bill, and her doctor Chris, have each selected a tentative morning agenda (from 8:00 AM to noon) and have each tasked a personal computational scheduling agent with maintaining schedules that can accomplish his/her agenda. Ann will have a 60 minute recreational activity (R^A) with Bill before spending 90 to 120 minutes performing a physical therapy regimen to help rehabilitate an injured knee (TR^A) (after receiving the prescription left by her doctor Chris); Bill will spend 60 minutes recreating (R^B) with Ann before spending 60 to 180 minutes at work (W^B); and finally, Chris will spend 90-120 minutes planning a physical therapy regimen (TP^C) for Ann and drop it off before giving a lecture (L^C) from 10:00 to 12:00. This example is displayed graphically as a distance graph (explained in Section 2.2.1) in Figure 2.1(a), with each event (e.g., the start time, ST, and end time, ET) appearing as a vertex, and constraints appearing as weighted edges.

One approach, displayed graphically in Figure 2.1(d), is for agents to simply select one joint schedule (an assignment of specific times to each event) from the set of

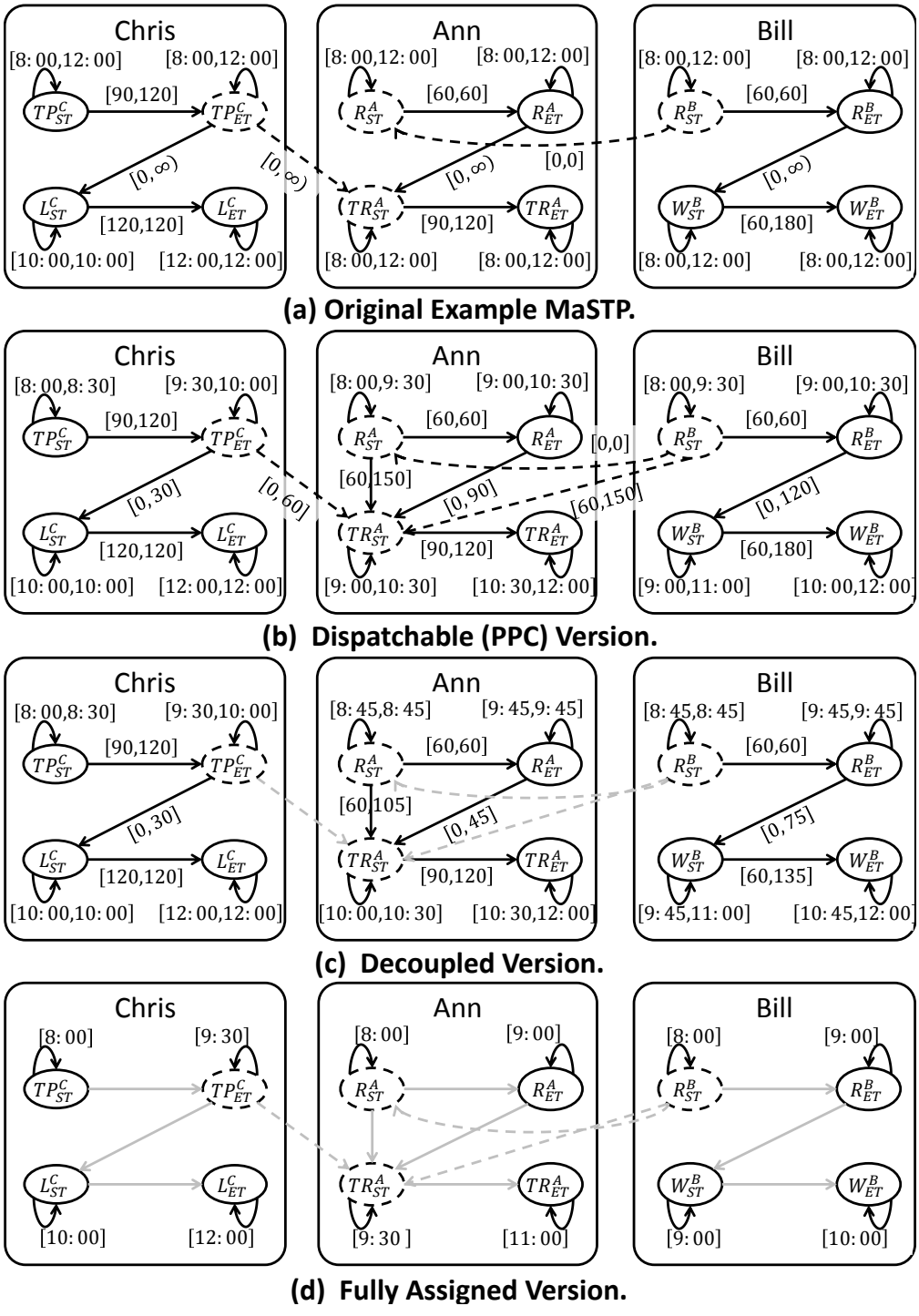

Figure 2.1: The distance graph corresponding to the (a) original, (b) minimal, (c) decoupled, and (d) fully assigned, versions of the example problem.

possible solutions. In this case, each agent can provide exact times when queried about possible timings and relationships between events, and do so confident that this advice will be independent of (and consistent with) the advice delivered by other agents. However, this also leads to agents offering very brittle advice, where, as soon as a new constraint arrives (e.g., Chris' bus arrives late by even a single minute), this exact, joint solution may no longer be valid. This, in turn, can require agents to regenerate a new solution *every* time a new constraint arrives, unless that new constraint happens to be consistent with the currently selected schedule.

A second approach for solving this problem is to represent the set of *all* possible joint schedules that satisfy the constraints, as displayed in Figure 2.1(b). This leads to advice that is much more robust to disturbances and accommodating to new constraints. In this approach, if a new constraint arrives (e.g., Chris' bus is late), the agents can easily recover by simply eliminating inconsistent joint schedules from consideration. However, doing so may still require communication (e.g., Chris' agent should communicate that her late start will impact when Ann can start therapy). The need for communication may continue (e.g., until Chris actually completes the prescription, Ann cannot start the therapy regimen), otherwise agents risk making inconsistent, concurrent decisions. For example, if both Ann's and Bill's agent report that recreation can start any time from 8:00 to 9:30 (all allowable times), but Ann decides to start at 8:00 while Bill decides to start at 9:00, the agents will have inadvertently given inaccurate (uncoordinated) advice, since Ann and Bill must start the recreation activity together.

There is also a third approach that attempts to balance the resiliency of Figure 2.1(b) with the independence of Figure 2.1(d). Agents can find and maintain a temporal decoupling, which is composed of *independent sets* of locally consistent schedules that, when combined, form a set of consistent joint schedules (Hunsberger, 2002). An example of a temporal decoupling is displayed in Figure 2.1(c), where, for example, Chris' agent has agreed to prescribe the therapy regimen by 10:00 and Ann's agent has agreed to wait to begin performing it until after 10:00. Then, not only will the agents' advice be independent of other agents', but it also provides agents with some resiliency to new constraints and offers users some flexibility and autonomy in making their own scheduling decisions. Now when Chris' bus is late by a minute, Chris' agent can absorb this new constraint by independently updating its local set of schedules, without requiring any communication with any other agent. The advantage of this approach is that once agents establish a temporal decoupling, there is no need for further communication unless (or until) a new (group of) constraint(s) render

the chosen decoupling inconsistent. It is only if and when a temporal decoupling does become inconsistent (e.g., Chris' bus is more than a half hour late, violating her commitment to finish the prescription by 10:00) that agents must calculate a new temporal decoupling (perhaps establishing a new hand-off deadline of 10:15), and then once again independently react to newly-arriving constraints, repeating the process as necessary.

Unfortunately, current solution algorithms (Dechter et al., 1991; Hunsberger, 2002; Xu & Choueiry, 2003; Planken et al., 2008b, 2010a) require centralizing the problem representation at some coordinator who calculates a (set of) solution schedule(s) for all. The computation, communication, and privacy costs associated with centralization may be unacceptable in multiagent planning and scheduling applications, such as military, health care, or disaster relief, where users specify problems to agents in a distributed fashion, and agents are expected to provide private, unilateral, time-critical, and coordinated scheduling assistance, to the extent possible.

In this chapter, I contribute new, *distributed* algorithms for finding both the complete joint solution space and temporal decouplings of the MaSTP. I prove the correctness and runtime properties of each these algorithms, including the level of independent, private reasoning that distributed algorithms can achieve. I also empirically compare the approaches, both with each other, to show the trade-offs in completeness *vs.* independence in reasoning, and with state-of-the-art centralized approaches, to show significant speedup over these approaches.

2.2 Background

In this section, I provide definitions necessary for understanding my contributions, using and extending terminology from the literature.

2.2.1 Simple Temporal Problem

As defined by Dechter et al. (1991), the Simple Temporal Problem (STP), $\mathcal{S} = \langle X, C \rangle$, consists of a set of timepoint variables, X, and a set of temporal difference constraints, C. Each timepoint variable represents an event and has a continuous domain of values (e.g., clock times) that can be expressed as a constraint relative to a special ***zero timepoint*** variable, $z \in V$, which represents the start of time. Each temporal difference constraint c_{ij} is of the form $x_j - x_i \leq b_{ij}$, where x_i and x_j are distinct timepoints, and $b_{ij} \in \mathbb{R}$ is a real number bound on the difference between x_j and x_i. Often, as notational convenience, two constraints, c_{ij} and c_{ji}, of the form

$b_{ji} \leq x_j - x_i \leq b_{ij}$ are represented as a single constraint of the form $x_j - x_i \in [-b_{ji}, b_{ij}]$.

A ***schedule*** is an assignment of specific time values to timepoint variables. An STP is ***consistent*** if it has at least one ***solution***, which is a schedule that respects all constraints.

	Availability	Duration	Ordering	External
Ann	$R^A_{ST} - z \in [480, 720]$	$R^A_{ET} - R^A_{ST} \in [60, 60]$	$R^A_{ET} - TR^A_{ST} \leq 0$	$R^A_{ST} - R^B_{ST} \in [0, 0]$
	$R^A_{ET} - z \in [480, 720]$			
	$TR^A_{ST} - z \in [480, 720]$	$TR^A_{ET} - TR^A_{ST} \in [90, 120]$		$TP^C_{ET} - TR^A_{ST} \leq 0$
	$TR^A_{ET} - z \in [480, 720]$			
Bill	$R^B_{ST} - z \in [480, 720]$	$R^B_{ET} - R^B_{ST} \in [60, 60]$	$R^B_{ET} - W^B_{ST} \leq 0$	$R^A_{ST} - R^B_{ST} \in [0, 0]$
	$R^B_{ET} - z \in [480, 720]$			
	$W^B_{ST} - z \in [480, 720]$	$W^B_{ET} - W^B_{ST} \in [60, 180]$		
	$W^B_{ET} - z \in [480, 720]$			
Chris	$TP^C_{ST} - z \in [480, 720]$	$TP^C_{ET} - TP^C_{ST} \in [90, 120]$	$TP^C_{ET} - L^C_{ST} \leq 0$	$TP^C_{ET} - TR^A_{ST} \leq 0$
	$TP^C_{ET} - z \in [480, 720]$			
	$L^C_{ST} - z \in [600, 600]$	$L^C_{ET} - L^C_{ST} \in [120, 120]$		
	$L^C_{ET} - z \in [720, 720]$			

Table 2.1: Summary of the running example problem.

In Table 2.1, I formalize the running example with specific constraints. Each activity has two timepoint variables representing its start time (ST) and end time (ET), respectively. In this example, all activities are to be scheduled in the morning (8:00-12:00), and so are constrained (Availability column) to take place within 480 and 720 of the zero timepoint z, which in this case is the start of day (midnight). Duration constraints are specified with bounds over the difference between an activity's end time and start time, whereas Ordering constraints dictate the order in which an agent's activities must take place with respect to each other. Finally, while a formal introduction of external constraints is deferred until later (Section 2.4), the last column represents constraints that span the subproblems of different agents. Figure 2.1 (d) illustrates a schedule that represents a solution to this particular problem.

2.2.2 Simple Temporal Network

To exploit extant graphical algorithms (e.g., shortest path algorithms) and efficiently reason over the constraints of an STP, each STP is associated with a Simple Temporal Network (STN), which can be represented by a weighted, directed graph, $\mathcal{G} = \langle V, E \rangle$, called a ***distance graph*** (Dechter & Pearl, 1987). The set of vertices V contains a vertex v_i for each timepoint variable $x_i \in X$, and E is a set of directed edges, where, for each constraint c_{ij} of the form $x_j - x_i \leq b_{ij}$, a directed edge, e_{ij} from v_i to v_j is constructed with an initial weight $w_{ij} = b_{ij}$. As a graphical short-hand, each