Artificial Intelligence 214 (2014) 26–44

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Flexibility and decoupling in Simple Temporal Networks ☆

Michel Wilson [a], Tomas Klos [a], Cees Witteveen [a,*], Bob Huisman [a,b]

[a] Department of Software and Computer Technology, TU Delft, Delft, The Netherlands
[b] Fleet Services, NedTrain Company, Utrecht, The Netherlands

ARTICLE INFO

Article history:
Received 20 December 2013
Received in revised form 9 May 2014
Accepted 10 May 2014
Available online 15 May 2014

Keywords:
Scheduling
Simple Temporal Network
Flexibility

ABSTRACT

We propose a new metric to determine the flexibility of a Simple Temporal Network (STN). After reviewing some existing flexibility metrics, we conclude that these metrics fail to capture the dependencies between events specified in the STN. As a consequence, these metrics will usually overestimate the available flexibility in such a system. We propose to use an intuitively more acceptable flexibility metric. This metric is based upon the notion of an interval schedule for an STN. Such an interval schedule specifies an interval for every event in the STN in such a way that, for every event, we are free to choose a starting time within its interval independently from the choice made for other events. We show that an interval schedule that maximizes our flexibility metric is computable in low-order polynomial time. As byproducts of this flexibility metric, we discuss simple solutions to problems in STNs with uncertainty (STNUs) and temporal decoupling in STNs. With respect to the latter we show that after computing our flexibility metric, we get a decomposition of the STN almost for free. Even more importantly, we show that contrary to popular belief, such a decomposition does not affect the flexibility of the original STN.

1. Introduction and background

A scheduling problem arises whenever there is a set of temporal events whose starting times are subject to a number of temporal (and maybe other) constraints. Then we have to find suitable time points for these starting times such that all constraints are satisfied. Scheduling problems occur in many diverse application domains such as transportation, process industry, health, and education [1–3]. A solution to a scheduling problem is called a *schedule* and is usually given in the form of an assignment σ of starting times $\sigma(e) = t_e$ to events e.

Although this problem has attracted quite some research and many interesting scheduling problems have been investigated, we are not primarily interested in proposing a single fixed-time schedule as a solution to a scheduling problem. The first reason for this is the limited *preservability* of fixed-time schedules as assignments determined off-line prior to execution. During execution of the schedule, unforeseen disturbances may occur that prevent certain events from starting at their predetermined times. Simply delaying such events is typically not an option, as this may in turn affect other events, and could ultimately violate a given deadline. Hence, even a single disturbance occurring during execution of a schedule could be a reason to modify it, or to compute another schedule from scratch. The problem with such on-line modifications is that we often do not have enough time to compute a new schedule when a disturbance occurs. Hence, it is better to be

☆ This paper is an invited revision of a Distinguished Paper at the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI-13).

* Corresponding author at: Faculty EEMCS, Mekelweg 4, 2628 CD Delft, The Netherlands. Tel.: +31 15 278 2521.
E-mail addresses: M.Wilson@tudelft.nl (M. Wilson), T.B.Klos@tudelft.nl (T. Klos), C.Witteveen@tudelft.nl (C. Witteveen), B.Huisman@tudelft.nl (B. Huisman).

http://dx.doi.org/10.1016/j.artint.2014.05.003

prepared for possible changes by using *flexible* schedules. There already exist quite a number of proposals for scheduling techniques that incorporate some form of flexibility in the final schedule [4,5]. Most of them, however, ensure flexibility in the off-line schedule-construction phase, but one still needs to apply a real-time transformation of these schedules in order to ensure their efficient execution [6]. In order to avoid such a time-critical transformation process we propose to compute an *interval schedule*, that is, an assignment of *time intervals* to events, such that for every event e one can freely choose a specific time point t in its interval, without risking any violation of scheduling constraints. Intuitively, such an interval for event e indicates the *flexibility* we have with respect to scheduling e.

Another reason to look for alternatives to fixed-time schedules is the presence of temporal events that are not under direct control of the scheduling agent (see e.g. [7–9]). It may be known only that starting times and/or ending times of these events take place in a given interval; the exact times are unknown or hard to predict. Even if we would come up with an exact schedule, it would only be a *partial assignment* of time points to events: We have to make sure that the schedule will not violate constraints, for any realization of the events not under our control. Here, too, it is quite natural to assign time intervals to events not under our control.

Fortunately, we are not the first ones to propose intervals instead of fixed times for scheduling events. For example, some authors (see e.g. [10,11,4,9]) already pointed out that if a scheduling formalism enjoys *decomposability*, we can use this property to achieve some form of flexibility by assigning an interval of time points to a certain event. Here, decomposability refers to the property of a scheduling system that for every partial schedule that satisfies the scheduling constraints, there is always a total schedule extending it that will satisfy the constraints. In particular, it implies that if for a certain event e there is an interval I_e of time points v_e that can be chosen as possible starting times for e, we are sure to be able to extend the partial schedule $\sigma(e) = v_e$ to a total schedule. Since this idea can be used starting from any event e, some authors have suggested to use these intervals I_e of time points to achieve a flexible schedule for such a system. We will discuss this idea in more detail below. Here, it suffices to remark that this intuitively appealing notion of decomposability is not sufficient to construct acceptable flexible schedules, for the reason that decomposability does not imply that given intervals are independent: Although decomposability allows one to determine the flexibility of two individual events taken separately, it does not allow us to infer the flexibility of these two events *simultaneously*.

We concentrate on the Simple Temporal Network (STN) [10,11], as the main representation mechanism for (temporal) scheduling problems, and we propose a new method for determining the flexibility inherent in an STN. Our approach enables us to find scheduling intervals I_e for the events e in an STN, in such a way that these intervals are *independent*: for every event e, I_e indicates the choice for a time point to be selected without any consequence for the choice of time points for other events. This notion of independence allows us to define a flexibility metric for STNs by taking the sum of the sizes of the individual intervals determined. As we will show, a metric based on this notion of flexibility can be computed efficiently and avoids some of the disadvantages of existing metrics, which are mainly based on the above mentioned notion of decomposability.

Our approach to flexibility of scheduling systems has a bearing on problems in *scheduling with uncertainty* and *distributed scheduling*, too. For scheduling with uncertainty, we concentrate on Simple Temporal Networks with Uncertainty (STNU) [12,8,13] and show that our approach to flexibility can be used in a very simple and straightforward way to solve the so-called *strong controllability problem* in STNU's.

In distributed scheduling we have a collection of constrained events, where these events are distributed over *multiple actors*. Typically, each such actor would like to schedule its events independently of the others. The problem of course is that each of these schedules, although locally consistent, can easily violate constraints pertaining to events that belong to different actors. In such cases we have to *decompose* the scheduling problem into subproblems for the individual agents in such a way that whatever local schedule is constructed by each of the individual actors, the merging of all these individual schedules results in a feasible solution of the total scheduling problem. One method that has been applied to fairly general scheduling formalisms is known as *temporal decoupling* [14,15]. Here, the original scheduling problem is decomposed into a number of subproblems, such that any solution (i.e., schedule) to a subproblem can be merged with any solution of the other subproblems, to constitute a feasible schedule for the total problem. As has been pointed out by various authors [14–18], when using existing flexibility metrics, applying temporal decoupling may seriously affect the total flexibility of the system. The question then arises, how we could minimize this flexibility loss due to temporal decoupling. As we will prove, using our flexibility metric, we can provide an optimal solution to this decoupling problem and – contrary to popular belief – even show that temporal decoupling does not imply any loss of flexibility of the original system.

Our approach to address these problems is the following. After a discussion of existing metrics for characterizing the flexibility of STNs, we show that these metrics are not able to capture *dependencies* between events, and generally overestimate the flexibility of scheduling systems. After a discussion of some rationality postulates for flexibility of an STN, we introduce a new flexibility metric using a transformation from a given STN S into a new STN S' such that dependencies between time points are resolved in S'. By using an LP-approach to find independent intervals for schedules in S', we obtain the flexibility of the original system S. We show that in case of STNs this flexibility metric can be computed in polynomial time and offers the possibility to adapt existing schedules in $O(1)$ time. Next, we discuss the strong controllability problem in STNUs and we show how to solve some generalized versions of this problem in a straightforward way. Finally, we deal with the problem of finding an optimal temporal decoupling for a given STN. We show that the method we use to specify the flexibility of an STN allows us to compute a decoupling for an STN. This decoupling is optimal: it does not induce any loss of flexibility due to decoupling and can be obtained in $O(k)$-time (if the flexibility of the system has already been