

Preference Propagation in Temporal/Capacity Constraint Graphs

Norman Sadeh and Mark S. Fox

CMU-RI-TR-89-2

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

January 5, 1989

Copyright © 1989 Sadeh&Fox

Also appears as Computer Science technical report CMU-CS-88-193.

This research has been supported, in part, by the Defense Advance Research Projects Agency under contract #F30602-88-C-0001, and in part by a grant from McDonnell Aircraft Company.

Table of Contents

1. Introduction	1
1.1. The Issue	1
1.2. Formalization of the Scheduling Problem	2
1.3. An Example	4
1.4. Organization of the Paper	7
2. Related Work	8
3. A Probabilistic Framework for Preference Propagation	11
3.1. An Overview of the Approach	11
3.2. Building A Priori Probability Distributions Based on Local A Priori Preferences	12
4. Propagating Start Time and Duration Distributions in a TCG	15
4.1. Preliminary Remarks	15
4.2. Propagation in an Acyclic TCG with Fixed-duration Activities	17
4.2.1. $C_i^0: I_0$ MEETS I_i^0	19
4.2.2. $C_i^0: I_0$ is BEFORE I_i^0	20
4.2.3. Example	21
4.3. Propagation in an Acyclic TCG with Variable-duration Activities	22
4.3.1. $C_i^0: I_0$ MEETS I_i^0	23
4.3.2. $C_i^0: I_0$ BEFORE I_i^0	23
4.4. Relaxing the Acyclicity Assumption	23
4.5. Propagation in a TCG with Explicit Disjunctions	32
4.6. Result Interpretation	33
5. Resource Demand Densities	34
6. Discussion	37
6.1. Time Complexity	37
6.2. Expressiveness of the Model	37
6.3. Possible Improvements	38
6.3.1. Iterative and Hierarchical Preference Propagation	38
6.3.2. Activity Criticality	38
6.3.3. Value Goodness	39
7. Summary and Concluding Remarks	39
Appendix 1: A Posteriori Start Time and Duration Distributions	41
1. Acyclic TCG with fixed-duration activities	41
1.1. $C_i^0: I_0$ MEETS I_i^0	41
1.2. $C_i^0: I_0$ MET-BY I_i^0	41
1.3. $C_i^0: I_0$ BEFORE I_i^0	41
1.4. $C_i^0: I_0$ AFTER I_i^0	41
1.5. $C_i^0: I_0$ DURING I_i^0	41
1.6. $C_i^0: I_0$ CONTAINS I_i^0	42
1.7. $C_i^0: I_0$ STARTS I_i^0	42
1.8. $C_i^0: I_0$ STARTED-BY I_i^0	42
1.9. $C_i^0: I_0$ FINISHES I_i^0	42
1.10. $C_i^0: I_0$ FINISHED-BY I_i^0	42
1.11. $C_i^0: I_0$ OVERLAPS I_i^0	42
1.12. $C_i^0: I_0$ OVERLAPPED-BY I_i^0	42
1.13. $C_i^0: I_0$ EQUALS I_i^0	43

2. Acyclic TCG with variable-duration activities	43
2.1. $C_i^0: I_0$ MEETS I_i^0	43
2.2. $C_i^0: I_0$ MET-BY I_i^0	43
2.3. $C_i^0: I_0$ BEFORE I_i^0	43
2.4. $C_i^0: I_0$ AFTER I_i^0	44
2.5. $C_i^0: I_0$ DURING I_i^0	44
2.6. $C_i^0: I_0$ CONTAINS I_i^0	44
2.7. $C_i^0: I_0$ STARTS I_i^0	44
2.8. $C_i^0: I_0$ STARTED-BY I_i^0	44
2.9. $C_i^0: I_0$ FINISHES I_i^0	44
2.10. $C_i^0: I_0$ FINISHED-BY I_i^0	45
2.11. $C_i^0: I_0$ OVERLAPS I_i^0	45
2.12. $C_i^0: I_0$ OVERLAPPED-BY I_i^0	45
2.13. $C_i^0: I_0$ EQUALS I_i^0	45
Appendix2: Activity Individual Demand Densities	46
1. Notations	46
2. Resource Demand Densities Produced by Fixed-Duration Activities	46
3. Resource Demand Densities Produced by Variable-Duration Activities	48
Acknowledgement	48

List of Figures

Figure 1-1:	Allen's 13 temporal relation constraints	3
Figure 1-2:	TCG for a two order scheduling problem	5
Figure 1-3:	Start time utility functions	6
Figure 1-4:	Gantt chart for a schedule of order1 obtained with scenario1. The global start time utility has been obtained by adding the start time utility of the five activities.	7
Figure 1-5:	Gantt chart for a schedule of order1 obtained with scenario2. The global start time utility has been obtained by adding the start time utility of the five activities.	7
Figure 2-1:	An inconsistent CSP with a consistent TCG	9
Figure 2-2:	A utility interpretation of DEVISER's start time windows	9
Figure 2-3:	Rit's generalized window	10
Figure 3-1:	A priori probability density $P(x)$ for a variable x with utility function $u(x)$. This is an example with a single optimal value.	14
Figure 4-1:	Example of a TCG with no explicit disjunctions	15
Figure 4-2:	Example of a TCG with explicit disjunctions	16
Figure 4-3:	An Acyclic TCG. The time interval I_0 is related to $I_1^0, I_2^0, \dots, I_{p_0}^0$ by respectively $C_1^0, C_2^0, \dots, C_{p_0}^0$.	18
Figure 4-4:	I_0 MEETS I_i^0	20
Figure 4-5:	I_0 BEFORE I_i^0	20
Figure 4-6:	A posteriori start time densities for order ₂	22
Figure 4-7:	Illustration of Fubini's Theorem in a TCG with 3 Time Periods	25
Figure 4-8:	A TCG with 3 Time Periods	26
Figure 4-9:	Procedure to express $P(C_1, \dots, C_m st_0 = t \& du_0 = d)$ as an iterated integral	28
Figure 4-10:	Main steps involved in the construction of (15). Notice that the α expressions have been omitted as they trivially evaluate to 1.	30
Figure 4-11:	A posteriori start time densities for order ₁	31
Figure 4-12:	A TCG with explicit disjunctions	32
Figure 5-1:	$R_1, R_2,$ and R_3 's aggregate demand densities	35
Figure 5-2:	Contributions of $A_3, A_4,$ and A_7 to R_2 's aggregate demand density	36

Abstract

Scheduling can be formalized as a constraint satisfaction problem (CSP). Within this framework activities in a plan are interconnected via temporal relation constraints a la Allen, thereby defining a temporal constraint graph (TCG). Additionally there are capacity constraints restricting the use of each resource to only one activity at a time. Together these constraints form a *temporal/capacity constraint graph* (T/CCG) . Preferences such as meeting due dates, reducing order flowtime, or selecting accurate machines are modeled as utility functions over the domain of possible start times and durations of activities and over the sets of possible resources activities can use. These preferences interact via the TCG and via the resource capacity constraints. Hence, in general, they cannot be simultaneously optimized. The objective of *preference propagation* techniques is to transform such local a priori preferences so as to account for their interactions.

In this paper we describe a probabilistic framework in which start time, duration and resource preferences are propagated across T/CCGs in order to focus attention in an *incremental scheduler*. The propagation is first performed across the TCG, thereby producing activity (a posteriori) start time and duration distributions. These distributions allow for early detection of unsolvable CSPs, and provide measures of value goodness and variable looseness for activity start times and durations. In a second phase, these distributions are combined to predict the degree of *contention for each resource* and the *reliance of each activity on the possession of that resource*.

1. Introduction

1.1. The Issue

We are concerned with the issue of how to opportunistically focus an incremental scheduler's attention on the most critical decision points and the most promising decisions in order to reduce search and improve the quality of the resulting schedule. More specifically we are concerned with incremental constraint directed scheduling where the problem is defined as a set of variables and a set of constraints. Both variables and constraints are determined by the initial scheduling problem and the earlier decisions made by the scheduler. The interactions of the constraints determine the structure of the problem space. We characterize the problem space with a set of texture measures that are used to both identify critical decision points and select a decision at each of these points. The process of analyzing the current problem and generating new decisions (e.g. scheduling an operation) is repeated, thereby resulting in the incremental construction of a schedule.

Real-life scheduling problems are subject to a variety of preferences [Johnson 74, Fox 83, Ow 84, Smith 86] such as meeting due dates, reducing the number of machine set-ups, reducing inventory costs, using accurate and/or fast machines, making sure that some jobs are performed within a single work-shift, etc. Although these preferences are usually set independently to one another, they interact. For instance selection of a good start time for an activity (e.g. to meet a due date) may prevent the selection of an accurate machine for another operation or may prevent meeting another job's due date. For this reason, selecting operation start times or allocating resources based solely on local a priori preferences is likely to result in poor schedules. Preference propagation is meant to allow for the construction of measures that reflect preference interactions. These measures can then serve to guide the construction of a good *overall* schedule rather than a schedule that locally optimizes a subset of preferences.

We perform preference propagation within a probabilistic framework. We associate with each variable's value a probability that reflects the likelihood that this value results in a good schedule overall (*value goodness*). These probabilities are refined by being propagated across the problem constraints. Value goodness is a texture measure that helps selecting assignments for variables. Identification of critical variables (i.e. decision points) is performed using another texture measure, called *variable looseness*. A critical variable or group of variables is one whose good overall values are likely to become unavailable if one were to start assigning values to other variables first. Notice that if our measures of value goodness were perfect, the order in which variables are instantiated would not matter. However such perfect measures could only be obtained by first solving the problem. Because in practice measures of value goodness contain some uncertainty, one has to account for the effects of assigning a value to a variable over the availability of good values for other variables. A variable instantiation order is accordingly defined starting with the most critical (i.e. least loose) variables. In this paper we are interested in the identification of critical activities (i.e. operations). An activity is made of a start time variable, possibly a duration variable, and a set of resource variables. We identify critical activities as the ones that heavily rely on the possession of highly contended resources. Indeed, if such critical activities are not scheduled first, it is very likely, that by the time the scheduler turns its attention to these activities, the resources that would have been the most appropriate for these activities will no longer be available.

We discuss preference propagation in temporal/capacity constraint graphs (T/CCG). Temporal constraints define partial orderings among the activities to be scheduled. All thirteen of Allen's [Allen 84] temporal relation constraints are accounted for. Resource capacity constraints restrict the use of resources to only one activity at a time. Both situations with fixed and variable duration activities are discussed. Our formalism allows for both activity start time and duration preferences as well as for resource preferences. It also accounts for prior resource reservations if any. It is shown that the (a posteriori) start time and duration distributions resulting from the propagation across the temporal constraints can be combined to identify resources that are highly contended for (*resource contention*) and activities that heavily rely on the possession of these resources (*activity resource reliance*) in function of time.

We also argue that a posteriori start time/duration distributions can be seen locally as measures of start time/duration goodness and globally as measures of start time/duration looseness. Our notion start time/duration looseness generalizes the Operations Research notion of slack [Baker 74, Johnson 74].

1.2. Formalization of the Scheduling Problem

The factory scheduling problem is often described as a two step problem: a *process planning* step and a *resource planning* step [Fox 83]. Process planning deals with the generation and selection of plans (i.e. process routings) that satisfy the order specifications. Resource planning, sometimes also referred to as scheduling, deals with the allocation of resources (e.g. machines) to activities and the assignment of start and end times to activities. In general both steps can be interleaved.

In this paper we will be concerned exclusively with the scheduling part of the problem: we will assume that we are given a set of plans to schedule. Here a plan is simply defined as a partial ordering of activities. Each activity may require one or more resources, for each of which there can be several alternatives.

We formalize the scheduling problem as a constraint satisfaction problem (CSP). The variables of the problem are the activity start times, the resources allocated to each activity, when there is a choice, and possibly the duration of each activity. An activity's end time is defined as the sum of the activity's start time and duration. We differentiate between two types of constraints: *required constraints* and *preferential constraints* [Fox 83]. Required constraints determine the admissibility of a solution to the CSP (schedule) while preferential constraints allow for differentiating among admissible solutions. The degree of satisfaction of a preferential constraint is defined by a utility function that maps the possible values of a variable onto utilities ranging between 0 and 1. A utility of 0 indicates a non-admissible value. A value with utility 1 is an optimal value.

We will be dealing explicitly with two types of required constraints: *temporal relation constraints* and *resource capacity constraints*. Temporal relation constraints are used to describe partial orderings among activities as provided by the process planning step. We will be using Allen's temporal relation constraints [Allen 84] to describe these constraints (Figure 1-1). We will refer to the graph defined by these constraints, for a given CSP, as the CSP's *temporal constraint graph* (TCG). Capacity constraints restrict the number of reservations of a resource over any time interval to the capacity of that resource. In this paper, for the sake of simplicity,

Relation -----	Pictorial Representation -----
X BEFORE Y Y AFTER X	XXX YYY
X EQUALS Y Y EQUALS X	XXX YYY
X MEETS Y Y MET-BY X	XXXYYY
X OVERLAPS Y Y OVERLAPPED-BY X	XXX YYY
X DURING Y Y CONTAINS X	XXX YYYYY
X STARTS Y Y STARTED-BY X	XXX YYYYY
X FINISHES Y Y FINISHED-BY X	XXX YYYYY

Figure 1-1: Allen's 13 temporal relation constraints

we will always be assuming resources with unary capacity. Together these required constraints form a temporal/capacity constraint graph (T/CCG). A schedule that does not satisfy the required constraints of the CSP is not admissible.

We will allow for preferential constraints on activity start times and durations as well as on the resources to be used by each activity. Preferential constraints are described with utility functions. For a given preferential constraint, a variable's value is admissible only if its utility is strictly positive. High preference for an admissible value is indicated by a high utility. In practice the domain of admissible values resulting from these preferential constraints, i.e. the domain with strictly positive utilities, is always bounded. For instance the domain of admissible start times of an activity is constrained at one end by the order release date and at the other end by the order due date according to the durations of the activities that precede/follow the activity within the plan.

Notations

We have to schedule a set of activities $\{A_1, A_2, \dots, A_n\}$. Let I_k denote the time interval over which A_k spans. st_k , et_k and du_k respectively denote I_k 's start time, end time, and duration. Activities are connected by a set of temporal relation constraints, thereby forming a TCG. We view TCGs as *undirected* graphs. An arc in a TCG indicates the presence of a temporal relation between two intervals (e.g. I_1 BEFORE I_2 or equivalently I_2 AFTER I_1). Let C_1, C_2, \dots, C_m denote the temporal relation constraints in the TCG. The TCG, which has been produced during the process planning phase, is assumed consistent.

Additionally there are capacity constraints limiting the use of each resource to only one activity at a time. By adding these capacity constraints to the TCG, one obtains the CSP's T/CCG. In this paper we will not need to formalize the description of T/CCGs any further as we will be mainly dealing with their temporal abstractions, i.e. the TCGs obtained by omitting the capacity constraints in the T/CCGs.

Each activity A_k has a preferential start time constraint with associated utility function denoted u_{st_k} . Activity A_k 's duration is either fixed or constrained by a preferential constraint with utility function u_{du_k} . The ranges of admissible start times and durations are assumed to be bounded, which is always the case in practice.

Each activity A_k may require one or more resources $R_{k1}, R_{k2}, \dots, R_{kp}$. For each resource R_{ki} required by activity A_k , there is a set of possible resources $R_{ki1}, R_{ki2}, \dots, R_{kij}$ available on the factory floor. This set is assumed to be finite, which is also always the case in practice. These different resources are usually not equally preferred. A resource utility function, $u_{R_{ki}}$ associates a utility (preference) $u_{R_{ki}}(R_{kij})$ with each possible resource R_{kij} . For instance a milling operation may require a milling machine and a human operator. There may be two milling machines available on the factory floor. For this specific milling operation, milling-machine₁ may have a preference of 1.0 and milling machine₂ a preference of 0.4 (e.g. due to a difference in the accuracy of the machines.). There may also be several human operators available with different utilities.

The global utility of a schedule is obtained by summing all the preferential constraints' utilities (for the given schedule).

1.3. An Example

We now introduce a simple scheduling problem that we will use throughout this paper.

The problem involves scheduling two orders: order₁ and order₂ (Figure 1-2):

- order₁ comprises five activities: A_1, A_2, \dots, A_5 ,
- order₂ comprises three activities: A_6, A_7 , and A_8 .

All activities have the same duration, namely 30 time units. C_1, C_2, \dots, C_7 are the temporal relation constraints imposed by the process planning step. For instance, C_1 indicates that A_1 has to precede A_2 . The domain comprises three physical resources: R_1, R_2 , and R_3 .

- A_1 requires a resource R_{11} which can be either R_1 or R_2 (with equal preference), i.e. $u_{R_{11}}(R_1)=u_{R_{11}}(R_2)=1$, and $u_{R_{11}}(R_3)=0$.
- A_2 requires a resource R_{21} which has to be R_1 , i.e. $u_{R_{21}}(R_1)=1$, and $u_{R_{21}}(R_2)=u_{R_{21}}(R_3)=0$.
- A_3 requires a resource R_{31} which has to be R_2 , i.e. $u_{R_{31}}(R_2)=1$, and $u_{R_{31}}(R_1)=u_{R_{31}}(R_3)=0$.
- A_4 requires a resource R_{41} which has to be R_2 , i.e. $u_{R_{41}}(R_2)=1$, and $u_{R_{41}}(R_1)=u_{R_{41}}(R_3)=0$.

- A_5 requires a resource R_{51} which has to be R_1 , i.e. $u_{R_{51}}(R_1)=1$, and $u_{R_{51}}(R_2)=u_{R_{51}}(R_3)=0$.
- A_6 requires a resource R_{61} which has to be R_3 , i.e. $u_{R_{61}}(R_3)=1$, and $u_{R_{61}}(R_1)=u_{R_{61}}(R_2)=0$.
- A_7 requires a resource R_{71} which can be either R_2 or R_3 (with equal preference), i.e. $u_{R_{71}}(R_2)=u_{R_{71}}(R_3)=1$, and $u_{R_{71}}(R_1)=0$.
- A_8 requires a resource R_{81} which has to be R_3 , i.e. $u_{R_{81}}(R_3)=1$, and $u_{R_{81}}(R_1)=u_{R_{81}}(R_2)=0$.

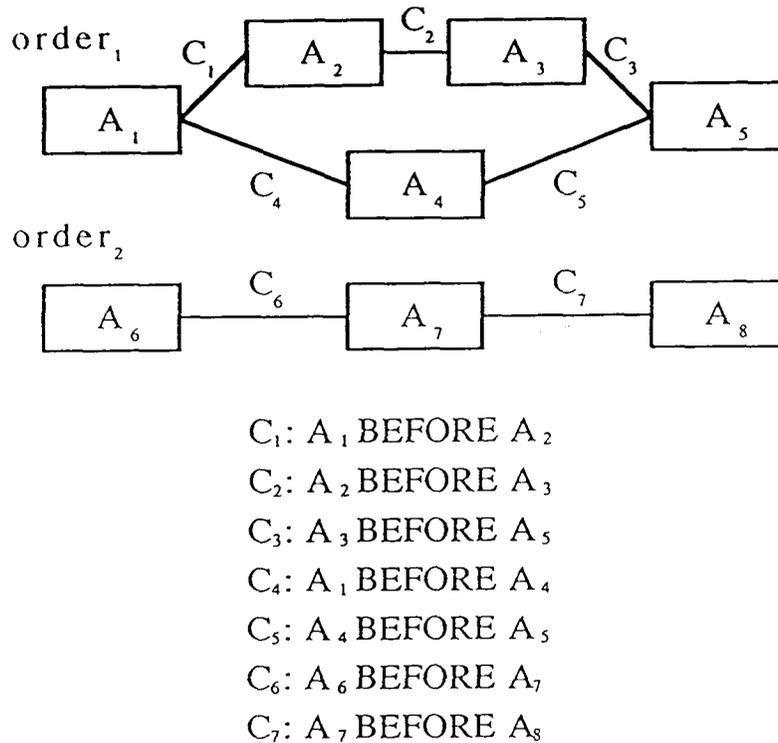


Figure 1-2: TCG for a two order scheduling problem

Activities in order₁ (i.e. A_1, \dots, A_5) are assumed to have the same start time utility function. The function requires that these activities start between time 0 and time 140, with an optimal start time at 120 (Figure 1-3). Time 0 can be interpreted as the order release date. Time $140 + 30 = 170$ (latest-start-time + duration = latest-finish-time) could represent the time after which the client would refuse the order.

All activities in order₂ will be assumed to have a uniform start time utility between 0 and 120 (Figure 1-3).

In general preferential constraints are set independently to one another and may therefore be incompatible. For instance, it will obviously be impossible to simultaneously schedule both A_2

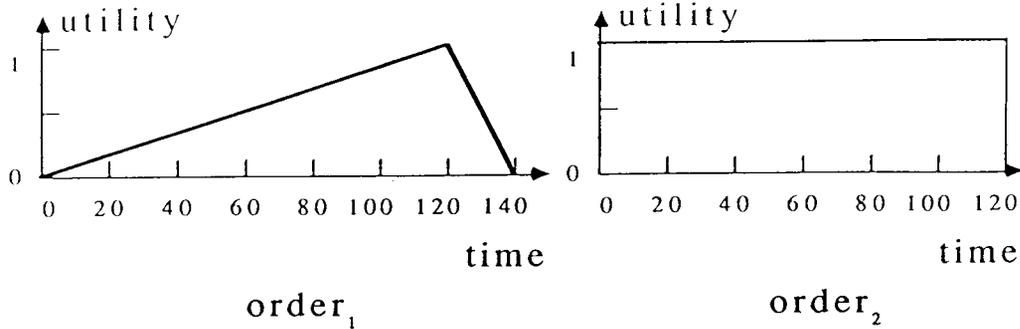


Figure 1-3: Start time utility functions

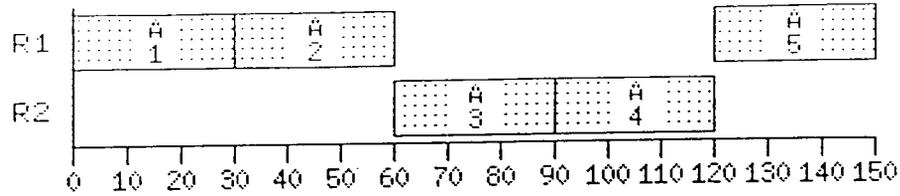
and A_3 at their optimal start time, namely 120. Therefore instead of a priori preferences, one needs (a posteriori) preferences indicating values that are likely to result in a good schedule *overall*. Such preferences can only be obtained by accounting for constraint interactions. This is the objective of the propagation technique presented in this paper.

Looking more closely at our example, one notices that there are four activities requiring resource R_2 : A_1 , A_3 , A_4 , and A_7 . Out of these four activities, up to three can occur in parallel, namely A_3 , A_4 , and A_7 . These activities will therefore compete for the possession of R_2 . There is no such competition for R_1 and R_3 as the activities that require these resources are fully ordered temporally (e.g. A_1 has to be carried out before A_2). The scheduling of R_2 is therefore more critical than that of R_1 and R_3 . An incremental scheduler should first focus its attention on the competition between A_3 , A_4 , and A_7 for R_2 . Moreover, since A_7 has two resource alternatives (R_2 and R_3) while A_3 and A_4 have only one, and since A_3 has less slack than A_4 , we would like our incremental scheduler to first schedule A_3 with R_2 ¹.

A more detailed analysis confirms that first scheduling A_3 rather than A_4 is the right decision. It also reveals the influence of the preferential constraints (in this case the start time preferences) in determining activity criticality. We consider two scenarios: one where the first activity to be scheduled is A_4 (**scenario1**) and one where it is A_3 (**scenario2**). Given that A_5 cannot start later than 140 and that A_3 and A_4 have a duration of 30, A_1 , A_2 , A_3 , and A_4 cannot start later than 110. Hence, according to their start time utility functions, these activities will prefer to start as late as possible (Figure 1-3). In **scenario1**, A_4 is the first activity to be scheduled. It is scheduled as late as possible (while still leaving some room for A_5 to have a good schedule), say at time 90. Since both A_3 and A_4 require R_2 , A_3 has to be scheduled before A_4 . The resulting schedule is the sequence A_1 , A_2 , A_3 , A_4 , and A_5 , as displayed in Figure 1-4. Alternatively (**scenario2**) suppose that we decide to first schedule A_3 . This time A_3 is scheduled to start at time 90, and A_4 has to occur before A_3 . However A_2 and A_4 can occur in parallel before A_3 (Figure 1-5). Hence, globally, A_1 to A_4 start later in **scenario2** than in **scenario1**. In other words, **scenario2** results in a higher global utility than **scenario1**.

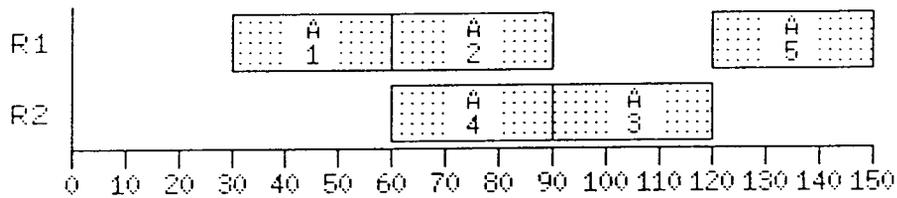
¹Notice that we are assuming an incremental scheduler whose reservations are nonpreemptible. The order in which activities are allocated resources would not matter if allocations were preemptible. Most predictive schedulers do not allow for such preemptions as they tend to produce infinite loops if one does not take special precaution.

Our approach to preference propagation formalizes the above considerations.



$$u_{st_1}(0) + u_{st_2}(30) + u_{st_3}(60) + u_{st_4}(90) + u_{st_5}(120) = 2.5$$

Figure 1-4: Gantt chart for a schedule of order1 obtained with **scenario1**.
The global start time utility has been obtained by adding the start time utility of the five activities.



$$u_{st_1}(30) + u_{st_2}(60) + u_{st_3}(90) + u_{st_4}(60) + u_{st_5}(120) = 3.0$$

Figure 1-5: Gantt chart for a schedule of order1 obtained with **scenario2**.
The global start time utility has been obtained by adding the start time utility of the five activities.

1.4. Organization of the Paper

In the next section we give an overview of related work in constraint satisfaction and scheduling. Section 3 introduces the assumptions that are the basis to our probabilistic approach to preference propagation and gives an overview of the approach. Section 4 describes the propagation of start time and duration probability distributions in TCGs. As already mentioned earlier, when one is simply concerned with knowing whether two time intervals are temporally related or not, a TCG can be considered as an undirected graph. *When we will be talking about cycles in the TCG, we will always be referring to the undirected interpretation of the graph.* Subsection 4.2 deals with acyclic TCGs with fixed-duration activities. Subsection 4.3 relaxes the fixed-duration hypothesis. Subsection 4.4 relaxes the acyclicity assumption. Subsection 4.5 discusses propagation in general TCGs where there are explicit disjunctions of temporal relation

constraints. Finally subsection 4.6 analyzes the results of the section with respect to a set of requirements and desiderata identified in subsection 3.2. Section 5 explains how the results of the previous section can be combined to obtain measures of resource contention and activity resource reliance. In section 6, we discuss the time complexity and the expressiveness of our framework as well as possible improvements. Section 7 summarizes the main ideas of the paper.

For the sake of concision, subsections 4.2 and 4.3 contain only the treatment of two temporal relation constraints (BEFORE, and MEETS). Formulas for the complete set of Allen's temporal relation constraints is presented in appendix 1. For the same reason, section 5 only sketches the computation of resource demand densities. The reader will find a complete treatment of these densities in appendix 2.

2. Related Work

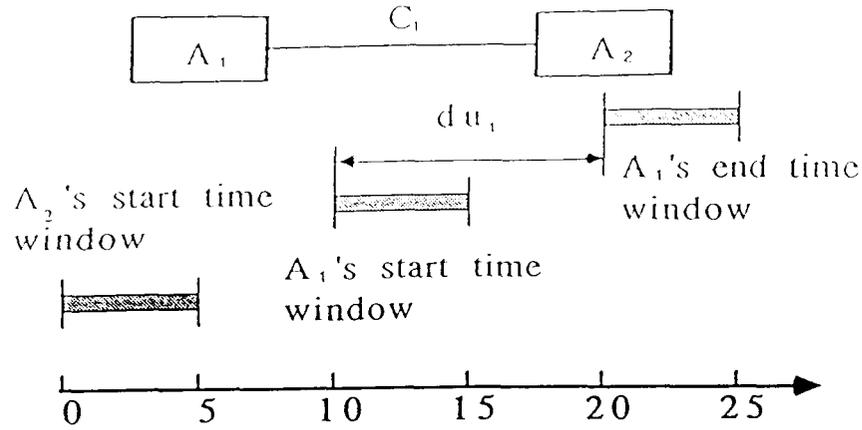
We already mentioned that this paper does not deal with process planning. Hence it is assumed that the TCGs to be scheduled are consistent. [Vilain 86] has proved that consistency checking in a general TCG is NP-hard. Several algorithms have been proposed in the literature to perform partial or total consistency checking in a TCG. Allen's algorithm [Allen 83] achieves 3-consistency² in a general TCG in polynomial time and space. A complete consistency checking algorithm using a variation of data dependency backtracking is presented in [Valdes-Perez 87]. Although the algorithm is designed for quick pruning, its asymptotic complexity remains exponential. [Vilain 86] and [Tsang 87] point out that consistency checking can actually be performed in polynomial time provided that the TCG does not contain certain types of disjunctive relations such as "Interval₁ has to be *either* BEFORE *or* AFTER Interval₂".

When additional constraints such as capacity constraints, preferential start time constraints, preferential duration constraints, preferential resource constraints, or resource reservations are added to a consistent TCG, the resulting CSP may stop being consistent. These are the types of inconsistencies that we will be referring to later in this paper. Consider the simple example depicted in Figure 2-1. There are two activities: A₁ and A₂. A₁ is BEFORE A₂. A₁ has a preferential start time constraint specifying that it has to start between 10 and 15. A₂'s preferential start time constraint specifies that A₂ has to start between 0 and 5. A₁'s duration is 10. A₂'s duration does not matter. The resulting CSP is obviously inconsistent (unsatisfiable) as A₁'s earliest end time (10+10=20) is after A₂'s latest start time (5).

Propagation of activity start and end time windows (earliest/latest start and end times) dates back to the CPM algorithm [Johnson 74]. The PERT method generalizes CPM by allowing for uncertainty in activity durations. [Vere 83] adapted the CPM propagation techniques to a planning system called DEVISER. In DEVISER start time windows are described as triples of the form (*earliest-start-time*, *ideal-start-time*, *latest-start-time*). The ideal start time information is optional. A start time triple can be seen as a triangle-shaped³ start time utility function (Figure 2-2). When the ideal start time is omitted, the window can be interpreted as a rectangle-shaped

²According to [Freuder 82], a constraint graph is *k-consistent* if for any set of (k-1) variables, any consistent assignment of values to these (k-1) variables, and any k-th variable, there always exists a value for the k-th variable such that the k values taken together (i.e. for the (k-1)+1 variables) are consistent.

³This is not the only possible interpretation.

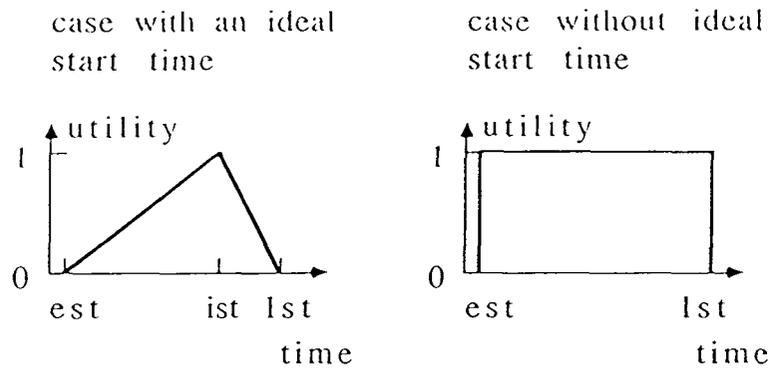


$C_1: A_1 \text{ BEFORE } A_2$

$du_1: A_1 \text{ 's duration}$

Figure 2-1: An inconsistent CSP with a consistent TCG

start time utility function. In DEVISER start time windows are dynamically compressed to account for new temporal relations and activities introduced during the planning process. The compression does not account for ideal start times which remain fixed. DEVISER is able to handle both BEFORE/AFTER and MEETS/MET-BY relations. Because preferences are not propagated, the system performs poorly when it has to select a start time within a compressed window: the selection is based on purely local information.



est: earliest start time

lst: latest start time

ist: ideal start time

Figure 2-2: A utility interpretation of DEVISER's start time windows

A variation of Vere's algorithm is presented in [Bell 84] that accounts for variable duration activities. [Smith 83] extends Vere's approach by also accounting for DURING/CONTAINS temporal relation constraints (see also [LePape 87]). Smith's temporal module also allows for the

propagation of resource reservations over T/CCGs. [Rit 86] describes a Waltz algorithm to propagate generalized temporal windows over TCGs. A generalized window is the composition of a start time, end time, and duration window (Figure 2-3). The method accounts for all 13 of Allen's temporal relations as well as for disjunctions among these relations. In the general case, Rit's algorithm is only guaranteed to achieve arc-consistency [Mackworth 77]. Total consistency can however be guaranteed in TCGs that do not contain disjunctions of temporal relation constraints.

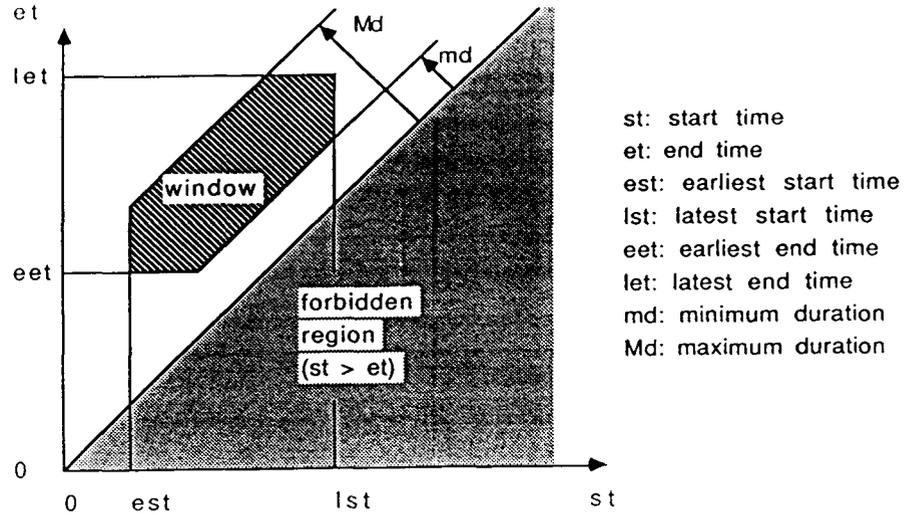


Figure 2-3: Rit's generalized window

None of the propagation techniques that we just described handles preferences. In practice, however, different times within a window are not equally preferred. For instance, in the factory scheduling domain, due dates and associated late delivery penalties induce preferences on activity end times (and hence start times). Inventory costs are another source of start time preferences. In general preferences cannot be accounted for independently. Selection of the optimal start time for one activity may prevent selection of the optimal start time for another either because of temporal relation constraints between the two activities or because of capacity constraints, or a combination of the two. This is why it is crucial not only to propagate time windows but also to propagate preferences over these windows. Window propagation simply guarantees admissibility of the values within a compressed window. *Preference propagation strives not only for admissibility but also for optimality by locally reflecting preference interactions.*

Our purpose is to develop preference propagation techniques to guide an incremental scheduler. Both empirical and analytical studies reported in [Haralick 80, Freuder 82, Purdom 83, Nadel 86a, Nadel 86b, Nadel 86c, Stone 86] indicate that, in general, the amount of search required to find a solution to a CSP can be significantly reduced by using the following two look-ahead schemes [Dechter 88]:

1. Variable Ordering: Focus on the most constrained variables first.
2. Value Ordering: Try the least constraining values first.

Tightly constrained variables and constraining values are determined by the interactions of the problem constraints. In CSPs where variables have finite sets of possible values and all values

are equally preferred, the number of possible values left after constraint propagation (i.e. consistency checking) can be used to determine variable tightness/looseness. In problems where values are not equally preferred, like in the scheduling domain, this is not sufficient. One has also to account for *value goodness*, i.e. the utility of a value and the impact of selecting that value on the availability of good values for the other variables. *Our notion of value goodness and our generalization of the notion of variable looseness are intended to allow for the generalization of these two look-ahead strategies to CSPs where variables can have infinite bounded sets of possible values with non-uniform preferences.*

[Muscettola 87] presents a probabilistic framework to compute resource contention in T/CCGs with only BEFORE/AFTER temporal relation constraints based on assumptions on the order in which the activities are scheduled. This paper extends Muscettola's approach for computing resource contention by removing the need for assumptions on the order in which activities are scheduled, by dealing with all of Allen's constraints, by allowing for duration and resource preferences and by accounting explicitly for earlier resource reservations.

3. A Probabilistic Framework for Preference Propagation

3.1. An Overview of the Approach

Our purpose is to develop preference propagation techniques to guide an incremental scheduler. An incremental scheduler works by iterating through a two-phase process. In the first phase it analyzes the structure of the CSP resulting from the initial scheduling problem and the decisions that have already been made. In the second phase, based on this analysis, new decisions are generated resulting in the expansion of the current schedule (e.g. new activities are scheduled). If the scheduler reaches a deadend, it backtracks. The process goes on until a satisfactory schedule is produced.

The first phase analysis is performed using preference propagation to dynamically identify critical decision points. Such decision points are determined by the interactions of the problem constraints. In the case of the scheduling problem, there are two main types of interactions: operation precedence interactions and resource requirement interactions [Smith 85].

1. The operation precedence interactions are the ones induced by the TCG. They are sometimes also referred to as *intra-order* interactions⁴.
2. Resource requirement interactions are induced by the capacity constraints. They arise from the contention of several activities for the same resource. They are sometimes referred to as *inter-order* interactions.

Intra-order and inter-order interactions have respectively motivated so-called order-based and resource-based scheduling techniques. In the past few years it has become clear that efficient scheduling requires the ability to combine these two perspectives [Smith 85, Smith 86] so as to account for both types of interactions. Unfortunately both types of interactions are not totally independent. Intra-order interactions affect the time intervals over which activities will contend for resources, thereby influencing inter-order interactions. Resource contention in turn restricts

⁴Although activities within a same order can also interact by competing for the same resources.

the times over which activities can occur, thereby influencing intra-order interactions.

In order to deal with the uncertainty in the interactions between uninstantiated variables, we have adopted a probabilistic model. For each uninstantiated variable, a probability density is computed for the variable's possible values that indicates the likelihood of each value to result in a good schedule overall, given the decisions already made by the incremental scheduler. This probability is a dynamic measure of value goodness. When the corresponding probability density is normalized, we also interpret this probability as the dynamic probability that the scheduler assigns that value to the variable.

In its simplest form our approach involves the following steps:

1. Based on the current partial schedule as well as start time, duration and resource preferences, *a priori* probability distributions are produced for the start time, duration and resources of each unscheduled activity,
2. These *a priori* probability distributions are propagated over the TCG, resulting in *a posteriori* start time and duration probability distributions,
3. The *a posteriori* distributions obtained in the previous step are combined to compute activity individual demand densities. An activity A_k 's *individual demand density* at time t for a resource R_{kij} , say $D_{kij}(t)$, is defined as the probability that A_k is active at time t and uses R_{kij} to fulfill its resource requirement R_{ki} ,
4. Activity individual demand densities are combined to measure resource *aggregate demand densities*. The aggregate demand density for a resource at time t is the probabilistic demand for that resource at time t .

Both iterative and hierarchical variations of this basic propagation algorithm will be discussed.

From an Operations Research point of view, activity *a posteriori* start time and durations provide a measure of intra-order interactions and a generalization of the notion of *slack* [Johnson 74]. Resource aggregate demand densities reflect the level of resource contention defined by the CSP's inter-order interactions. Resource aggregate demand densities can be identified with the Operations Research concept of *bottleneck analysis* [Smith 85, Smith 86, Muscettola 87].

From a constraint satisfaction perspective, a *a posteriori* start time/duration distributions should be regarded locally as measures of *value goodness*, and globally as measures of *variable looseness* for activity start times and durations. Aggregate demand densities can be interpreted as measures of *constraint contention* and individual demand densities as measures of *activity resource reliance*.

3.2. Building A Priori Probability Distributions Based on Local A Priori Preferences

Our approach uses Bayesian probabilities to estimate value goodness, i.e. the likelihood that a given value will result in a good schedule overall. It consists in the construction of *a priori* probability distributions for each uninstantiated variable based on local preferences. These probabilities are then refined so as to account for constraint interactions, thereby resulting into *a posteriori* probability distributions.

Obviously the main concern in such an approach is to obtain good estimates of value goodness

(**desideratum1**) as these estimates are essential to both the identification of critical decision points in the search space and the selection of a decision at these points. There are however some more specific requirements and desiderata to which one should give special attention. Indeed, besides its need for good focus of attention mechanisms, efficient search also requires the ability to quickly prune deadend paths in the search tree.

In our probabilistic framework, unsatisfiability is detected when a posteriori probability densities are uniformly zero. This indicates that the interactions of the problem constraints have reduced the set of admissible values for a variable to the empty set. Detecting unsatisfiability in this fashion requires that:

1. **Requirement1:** Every value that is a priori admissible⁵ is given a strictly positive a priori probability, though possibly very small.
2. **Requirement2:** The propagation step, which combines a priori probabilities to account for constraint interactions, produces a posteriori probability densities that are zero only for values forbidden by the constraint interactions.

Requirement1 restricts the construction of a priori probability distributions. **Requirement2** is a restriction on the propagation method itself.

Desideratum2: Additionally, in order to detect and prune inconsistent states as soon as possible, one would like a posteriori probabilities to be zero for all non-admissible values (complete consistency checking). Unfortunately interactions between the resource requirements of unscheduled activities seem computationally very expensive to totally account for. Consequently we will have to settle for partial consistency checking.

These general requirements and desiderata having been identified, we turn our attention to the construction of a priori probability distributions. We start with some general observations.

In the presence of a unique variable with a single (unary) preferential constraint, one can just select one of the optimal values defined by the utility function. The probability density for the variable's value consists of a set of peak distributions (Dirac distributions), each centered around one of the optimal values (Figure 3-1b).

On the other hand, in the presence of several variables and constraints, it is not always possible anymore to simultaneously select an optimal value for each variable. For instance, it is not always possible to schedule an activity at its optimal start time and with a set of optimal resources. Very often one has to settle for suboptimal start times and/or resources in order to find a feasible schedule (i.e. satisfy all the CSP's constraints). If, for a given variable, the interactions defined by the constraints are weak, it is usually possible to select a value that is still very close to the optimum (or one of the optimums). As interactions become stronger, it becomes more difficult to select values close to the optimums: the probability density widens (Figure 3-1c and d). In situations of extremely strong interactions, one is just happy to find a solution within the domain of admissibility (non-zero utility value). Hence the probability distribution tends towards a uniform distribution over the range of admissible values (Figure 3-1e).

⁵Of course, a more sophisticated analysis will result in the rejection of a larger number of possible values (see **desideratum2**). Therefore what is really important for **requirement1** is that *no* value received a zero a priori probability while it could have resulted in an admissible schedule.

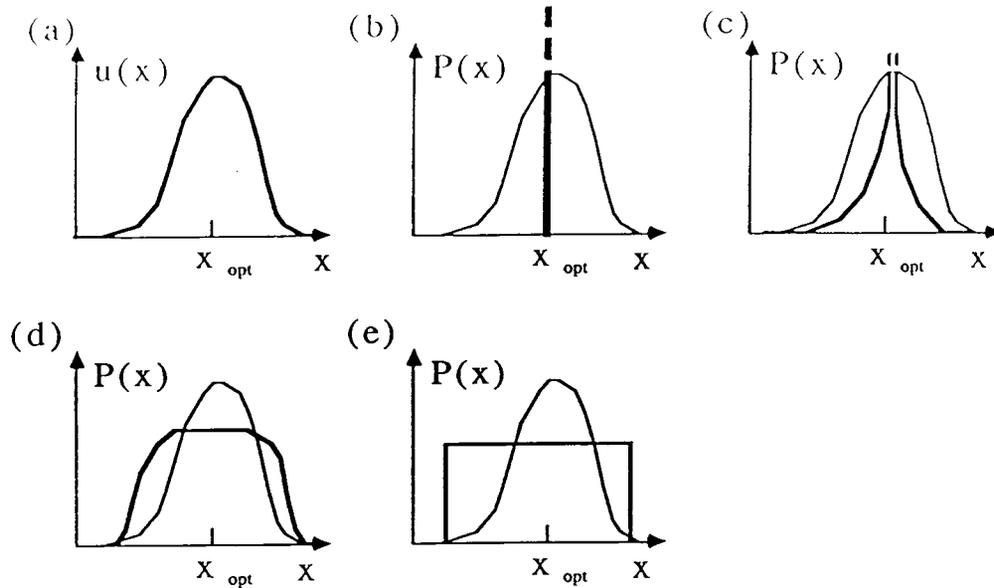


Figure 3-1: A priori probability density $P(x)$ for a variable x with utility function $u(x)$. This is an example with a single optimal value.

The a priori probability densities that we are currently using are essentially obtained by normalizing utility functions. Intuitively this can be interpreted as a sort of average difficulty assumption (see Figure 3-1). The resulting a priori probabilities obviously satisfy **requirement1**. Moreover, because we assume that domains of admissibility defined by utility functions are bounded, normalization is always possible.

Prior to normalizing a utility function, its domain is pruned to account for earlier resource reservations. This improves the quality of the probability distributions with respect to **desideratum2**. In particular we remove from the start time probability distributions the start times that are not allowed by the current resource reservations. A start time t is not allowed for an activity A_k , if there is at least one resource R_{ki} required by A_k such that none of the resources R_{kij} is totally available between t and $t + \delta_{k_{min}}$, where $\delta_{k_{min}}$ is A_k 's smallest admissible duration.

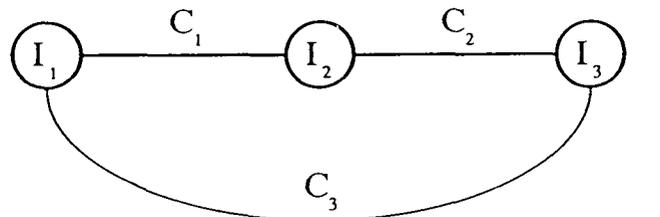
We are currently investigating alternative methods for producing a priori probability distributions. In particular we are investigating both iterative and hierarchical approaches to preference propagation. In an iterative approach one can use the resource demand densities obtained by the previous iteration to estimate the probability that a given resource will be available for an activity at some time t . Using these probabilities, new a priori start time probability distributions can be obtained and the propagation process can be carried out all over again. Alternatively, in a hierarchical scheme, one can use the propagation results obtained at an upper level to compute resource availability estimates. Again these estimates can be combined with the start time utility functions to obtain a priori start time probability distributions for the new level.

4. Propagating Start Time and Duration Distributions in a TCG

4.1. Preliminary Remarks

Now that we have some a priori distributions for the start time, duration (if variable) and resources of an activity, we can refine these a priori probabilities so as to account for the actual constraints of the problem. In this section we compute a posteriori start time (and duration) probabilities that account for the interactions defined by the TCG.

This section is subdivided into several subsections each dealing with the propagation problem under increasingly more general assumptions. As we already mentioned earlier we view TCGs as undirected graphs. Subsection 4.2 develops the computation of a posteriori start time probability distributions in acyclic TCGs with fixed duration activities. In subsection 4.3 the fixed-duration assumption is relaxed. In subsection 4.4 we relax the acyclicity assumption. Subsections 4.2 to 4.4 all assume that there are no explicit disjunctions in the TCG. The transitivity properties [Allen 83] of the relations may however induce disjunctions, which are then implicitly accounted for in the propagation. An example of TCG with no explicit disjunction is represented in Figure 4-1. The TCG specifies that I_1 OVERLAPS I_2 , and I_2 is DURING I_3 . This implicitly induces the disjunction $I_1 \{ \text{DURING, STARTS, OVERLAPS} \} I_3$, i.e. I_1 is DURING or STARTS or OVERLAPS I_3 . All our calculations allow for this type of implicit disjunctions. On the other hand explicit disjunctions are more difficult to handle and are quite infrequent in practical factory scheduling problems. Propagation in TCGs with explicit disjunctions is discussed in subsection 4.5. Figure 4-2 displays a TCG with explicit disjunctions. Subsection 4.6 interprets the results.



———— explicit constraint

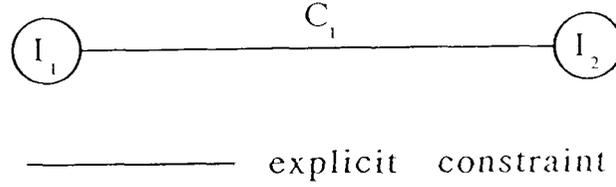
———— implicit constraint

$C_1: I_1 \text{ OVERLAPS } I_2$

$C_2: I_2 \text{ DURING } I_3$

$C_3: I_1 \{ \text{DURING, STARTS, OVERLAPS} \} I_3$

Figure 4-1: Example of a TCG with no explicit disjunctions



$$C_1: I_1\{\text{BEFORE, AFTER}\} I_2$$

Figure 4-2: Example of a TCG with explicit disjunctions

Notations

- C_1, C_2, \dots, C_m will denote the explicit temporal relation constraints that define the TCG (See Figure 4-1 for an example).
- $\sigma_k(st_k=t)$ will denote I_k 's a priori start time probability density, obtained as suggested in the previous section. t is the variable.
- $\delta_k(du_k=d)$ will denote I_k 's a priori duration probability density, obtained as suggested in the previous section. d is the variable. This distribution will only be used for variable-duration activities.
- $P(st_k=t \& C_1 \& C_2 \& \dots \& C_m)$ (where A_k is assumed to be a fixed-duration activity) will denote I_k 's a posteriori start time probability density. This density (with variable t) corresponds to the a posteriori probability that A_k starts at time t (i.e. $st_k=t$) and that the temporal relation constraints C_1, C_2, \dots, C_m are satisfied.
- $P(st_k=t \& du_k=d \& C_1 \& C_2 \& \dots \& C_m)$ (where A_k is assumed to be a variable-duration activity) will denote the two-dimensional joint a posteriori probability density of I_k 's start time and duration. This density (with variables t and d) corresponds to the a posteriori probability that A_k starts at time t (i.e. $st_k=t$) and has duration d (i.e. $du_k=d$) and that the temporal relation constraints C_1, C_2, \dots, C_m are all satisfied.

In order to avoid the accumulation of parentheses in iterated integrals, we adopt the usual convention that:

$$\int_A^B f(\xi) d\xi \int_{L(\xi)}^{U(\xi)} h(\eta) d\eta \text{ denotes } \int_A^B [f(\xi) \int_{L(\xi)}^{U(\xi)} h(\eta) d\eta] d\xi$$

We will also be using the following functions:

- $\alpha(\text{predicate})$ is a function that returns 1 when *predicate* evaluates to **true** and 0 otherwise⁶.

⁶The reader who is not familiar with this formalism can think of it as a convenient way of expressing IF-statements in mathematical formulas.

- $\beta^1[EQ_1(\xi)]$, where $EQ_1(\xi)$ is a linear equation in ξ , is a distribution⁷ such that:

$$\begin{aligned} \int_L^U \beta^1[EQ_1(\xi)] g(\xi) d\xi \\ &= \alpha[L < x < U] g(x) \text{ if } x \text{ is the unique solution to } EQ_1(\xi), \\ &= \int_L^U g(\xi) d\xi \text{ if } EQ_1(\xi) \text{ holds for } \forall \xi, \text{ and} \\ &= 0 \text{ if } EQ_1(\xi) \text{ is inconsistent.} \end{aligned}$$

This simply expresses that the integration variable ξ is not only restricted to values between L and U but that the values it can take should also satisfy the linear equation $EQ_1(\xi)$.

- More generally $\beta^l[EQ_1(\xi_1, \dots, \xi_n), \dots, EQ_l(\xi_1, \dots, \xi_n)]$, where $EQ_i(\xi_1, \dots, \xi_n)$ (for $i=1$ to l) is a linear equation, is a distribution such that:

$$\begin{aligned} \int_{L_1}^U d\xi_1 \int_{L_2}^U d\xi_2 \dots \int_{L_n}^U \beta^l[EQ_1(\xi_1, \dots, \xi_n), \dots, EQ_l(\xi_1, \dots, \xi_n)] g[\xi_1, \dots, \xi_n] d\xi_n \\ = \int_{L_k}^U d\xi_k \int_{L_{k+1}}^U d\xi_{k+1} \dots \int_{L_n}^U \alpha[L_1 < F^1(\xi_k, \xi_{k+1}, \dots, \xi_n) < U_1] \dots \alpha[L_{k-1} < F^{k-1}(\xi_k, \xi_{k+1}, \dots, \xi_n) < U_{k-1}] \\ \quad g[F^1(\xi_k, \xi_{k+1}, \dots, \xi_n), \dots, F^{k-1}(\xi_k, \xi_{k+1}, \dots, \xi_n), \xi_k, \xi_{k+1}, \dots, \xi_n] d\xi_n \end{aligned}$$

if the system of linear equations is consistent and equivalent to:

$$\begin{aligned} \xi_1 &= F^1(\xi_k, \xi_{k+1}, \dots, \xi_n) \\ \xi_2 &= F^2(\xi_k, \xi_{k+1}, \dots, \xi_n) \\ &\vdots \\ \xi_{k-1} &= F^{k-1}(\xi_k, \xi_{k+1}, \dots, \xi_n) \quad (k-1 \leq l) \end{aligned}$$

= 0 if the system of equations is inconsistent.

4.2. Propagation in an Acyclic TCG with Fixed-duration Activities

In this subsection we express the a posteriori probability density $P(st_0 = t \& C_1 \& C_2 \& \dots \& C_m)$ for the start time of an arbitrary time interval I_0 in terms of the a priori start time distributions⁸. We assume fixed-duration activities arranged in an acyclic TCG.

We denote by $I_1^0, I_2^0, \dots, I_{p_0}^0$ the intervals directly adjacent to I_0 in the TCG (Figure 4-3). C_i^0 ($1 \leq i \leq p_0$) is the temporal constraint between I_0 and I_i^0 . Each time interval I_i^0 ($1 \leq i \leq p_0$) is itself related directly or indirectly to some other time intervals by a set of constraints S_i^0 . The sets S_i^0 are disjoint as the TCG is assumed to be acyclic (Figure 4-3).

Since the TCG may be disconnected, which is the case when there are several independent orders to schedule, we have:

$$\{C_1^0\} \cup S_1^0 \cup \{C_2^0\} \cup S_2^0 \dots \cup \{C_{p_0}^0\} \cup S_{p_0}^0 \subseteq \{C_1, C_2, \dots, C_m\} \quad (1)$$

⁷Our β distribution is a variation of the Dirac distribution. The reader who is not familiar with this formalism can simply look at it as a convenient way of expressing constraints on the values that an integration variable can take.

⁸ I_0 is an arbitrary element of $\{I_1, I_2, \dots, I_n\}$.

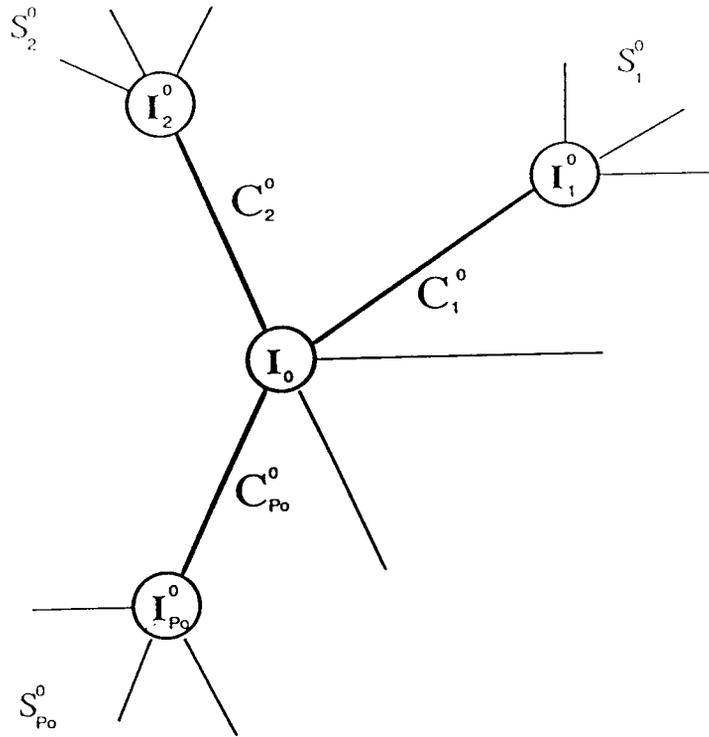


Figure 4-3: An Acyclic TCG.

The time interval I_0 is related to $I_1^0, I_2^0, \dots, I_{p_0}^0$
by respectively $C_1^0, C_2^0, \dots, C_{p_0}^0$.

and $m \leq n - 1$ (number of edges in an n -vertex tree).

We will express the (a posteriori) probability that I_0 starts at time t and that the constraints C_1, C_2, \dots, C_m are satisfied in terms of

- the a priori probability that I_0 starts at t , and
- the probabilities that each time interval I_i^0 ($1 \leq i \leq p_0$) has a start time compatible with C_i^0 given that $st_0=t$.

These latter probabilities can be expressed in a similar fashion, thereby resulting in an inductive formulation of the a posteriori start time probabilities. The inductive formulation process stops when all related time intervals have been accounted for (or more precisely their a priori start time distributions). At that point we have an expression of the a posteriori start time distribution that only contains a priori start time distributions, i.e. distributions that we know from the previous section.

Indeed, the a posteriori probability that $st_0=t$ and that the temporal relation constraints $C_1,$

C_2, \dots, C_m are satisfied is given by⁹ the a priori probability that I_0 starts at time t , denoted $\sigma_0(st_0=t)$, multiplied by the conditional probability that C_1, C_2, \dots, C_m are satisfied, given that $st_0=t$, denoted $P(C_1 \& C_2 \& \dots \& C_m | st_0 = t)$:

$$P(st_0 = t \& C_1 \& C_2 \& \dots \& C_m) = \sigma_0(st_0 = t) \times P(C_1 \& C_2 \& \dots \& C_m | st_0 = t) \quad (2)$$

with t being the distribution variable.

Furthermore, assuming I_0 's start time fixed at time t , the satisfaction of the constraints $\{C_i^0\} \cup S_i^0$ is independent of the satisfaction of the constraints $\{C_j^0\} \cup S_j^0$ (for $i \neq j$), since we are dealing with an acyclic TCG. Hence:

$$P(C_1 \& C_2 \& \dots \& C_m | st_0 = t) = \prod_{i=1}^{P_0} P(C_i^0 \& S_i^0 | st_0 = t) \quad (3)$$

where $P(C_i^0 \& S_i^0 | st_0 = t)$ is the conditional probability distribution that C_i^0 and the constraints in S_i^0 are satisfied given that $st_0 = t$ (with t being the distribution variable).

Using (3) we can now account separately for each constraint C_i^0 . We will express each multiplicand, $P(C_i^0 \& S_i^0 | st_0 = t)$, in terms of $P(st_i^0 = \tau \& S_i^0)$, the probability that I_i^0 starts at some time τ (to be defined) and that the constraints in S_i^0 are satisfied. Consequently equations (2) and (3) will enable us to express $P(st_0 = t \& C_1 \& C_2 \& \dots \& C_m)$ in terms of probabilities of the form $P(st_i^0 = \tau \& S_i^0)$, i.e. probabilities of the same form as the original probability $P(st_0 = t \& C_1 \& C_2 \& \dots \& C_m)$ except that S_i^0 is only a subset of $\{C_1, C_2, \dots, C_m\}$. By recursively repeating this process, we will be able to account for all the temporal relation constraints. The recursion process stops when S_i^0 gets empty since at that point $P(st_i^0 = \tau \& S_i^0) = P(st_i^0 = \tau) = \sigma_i^0(st_i^0 = \tau)$, I_i^0 's a priori start time density.

Paragraphs 4.2.1 and 4.2.2 develop the computation of $P(C_i^0 \& S_i^0 | st_0 = t)$ in terms of $P(st_i^0 = \tau \& S_i^0) = \sigma_i^0(st_i^0 = \tau)P(S_i^0 | st_i^0 = \tau)$ in the case where C_i^0 is respectively of the form " I_0 MEETS I_i^0 ", and " I_0 BEFORE I_i^0 ". The treatment of the set of all thirteen of Allen's temporal relation constraints can be found in appendix 1.

4.2.1. C_i^0 : I_0 MEETS I_i^0

The constraint " I_0 MEETS I_i^0 " requires (Figure 4-4) that I_0 's end time be equal to I_i^0 's start time, i.e. it requires that $et_0 = st_0 + du_0 = st_i^0$. In other words, assuming that $st_0 = t$, the probability that C_i^0 and the constraints in S_i^0 are satisfied is equal to the probability that $st_i^0 = t + du_0$ and that the constraints in S_i^0 are satisfied:

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= P(st_i^0 = t + du_0 \& S_i^0) \\ &= \sigma_i^0(st_i^0 = t + du_0) P(S_i^0 | st_i^0 = t + du_0) \end{aligned} \quad (4)$$

⁹ $P(A \& B) = P(A) \times P(B|A)$: the joint probability of two events A and B can be expressed as the product of the probability of event A with the conditional probability that B occurs given that A is assumed to occur.

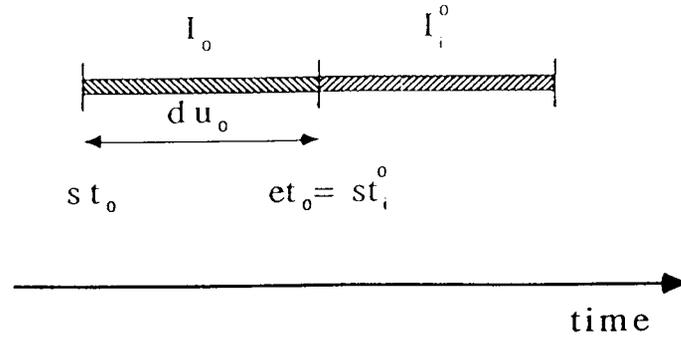


Figure 4-4: I_0 MEETS I_i^0

4.2.2. C_i^0 : I_0 is BEFORE I_i^0

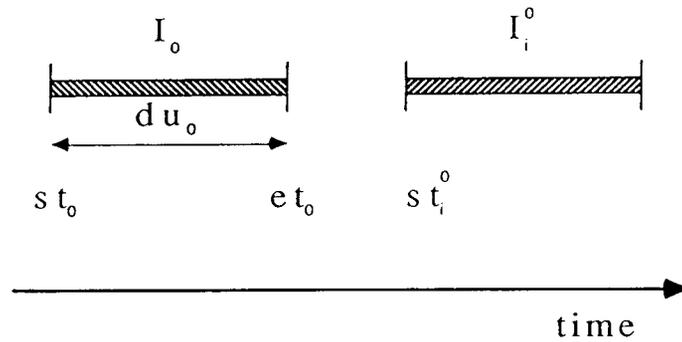


Figure 4-5: I_0 BEFORE I_i^0

The constraint " I_0 BEFORE I_i^0 " requires (Figure 4-5) that I_0 's end time be smaller than I_i^0 's start time, i.e. it requires that $et_0 = st_0 + du_0 < st_i^0$. In other words, assuming that $st_0 = t$, the probability that C_i^0 and the constraints in S_i^0 are satisfied is equal to the probability that $st_i^0 > t + du_0$ and that the constraints in S_i^0 are satisfied:

$$\begin{aligned}
 P(C_i^0 \& S_i^0 | st_0 = t) &= \int_{t + du_0}^{\infty} P(st_i^0 = \tau \& S_i^0) d\tau \\
 &= \int_{t + du_0}^{\infty} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau) d\tau
 \end{aligned} \tag{5}$$

4.2.3. Example

We use the three activities of order₂ (Figure 1-2) to illustrate the computations that we have just developed. We have:

$$\begin{aligned}
P(st_6=t \& C_1 \& C_2 \& \dots \& C_7) \\
&= P(st_6=t \& C_6 \& C_7) \\
&= \sigma_6(st_6=t) P(C_6 \& C_7 | st_6=t) \\
&= \sigma_6(st_6=t) \int_{t+du_6}^{\infty} \sigma_7(st_7=\tau_7) P(C_7 | st_7=\tau_7) d\tau_7 \\
&= \sigma_6(st_6=t) \int_{t+du_6}^{\infty} \sigma_7(st_7=\tau_7) d\tau_7 \int_{\tau_7+du_7}^{\infty} \sigma_8(st_8=\tau_8) d\tau_8
\end{aligned} \tag{6}$$

Also, using the formula given in appendix 1 to account for "I₇ AFTER I₆":

$$\begin{aligned}
P(st_7=t \& C_1 \& C_2 \& \dots \& C_7) \\
&= P(st_7=t \& C_6 \& C_7) \\
&= \sigma_7(st_7=t) P(C_6 \& C_7 | st_7=t) \\
&= \sigma_7(st_7=t) P(C_6 | st_7=t) P(C_7 | st_7=t) \\
&= \sigma_7(st_7=t) \int_{-\infty}^{t-du_6} \sigma_6(st_6=\tau_6) d\tau_6 \int_{t+du_7}^{\infty} \sigma_8(st_8=\tau_8) d\tau_8
\end{aligned} \tag{7}$$

Finally, in the same fashion:

$$\begin{aligned}
P(st_8=t \& C_1 \& C_2 \& \dots \& C_7) \\
&= P(st_8=t \& C_6 \& C_7) \\
&= \sigma_8(st_8=t) P(C_6 \& C_7 | st_8=t) \\
&= \sigma_8(st_8=t) \int_{-\infty}^{t-du_7} \sigma_7(st_7=\tau_7) P(C_6 | st_7=\tau_7) d\tau_7 \\
&= \sigma_8(st_8=t) \int_{-\infty}^{t-du_7} \sigma_7(st_7=\tau_7) d\tau_7 \int_{-\infty}^{\tau_7-du_6} \sigma_6(st_6=\tau_6) d\tau_6
\end{aligned} \tag{8}$$

We assume that the resources are initially free (i.e. no priori resource reservations). Therefore, since activities A₆, A₇, and A₈ have uniform start time utility functions, their a priori start time densities are uniform as well, and span between times 0 and 120. Figure 4-6 displays the a posteriori start time densities computed using these a priori densities. Since in this case the start time utilities are uniform, the most preferable start times for each activity are the ones that leave the most freedom to the other activities for satisfying the temporal constraints C₆ and C₇. For instance, A₆'s a posteriori start time density indicates that A₆ should start as early as possible in order to leave as much room as possible to A₇ and A₈¹⁰. As we will see in the example of subsection 4.4, the propagation of nonuniform start time utility functions such as the ones of the activities in order₁ are influenced by a second factor. In addition to looking for start times that leave a lot of slack to the other activities that have not been scheduled yet, the propagation of nonuniform utilities gives a higher preference to higher utilities. The a posteriori distributions

¹⁰It is important at this point to bear in mind that we have not yet accounted for resource capacity constraints. In particular we do not know the effects that these constraints will have on the domain of possible start times for A₇ and A₈. Hence, at this point, the best start times are the least committing ones with respect to the temporal constraints.

thereby reflect a compromise between the utilities to optimize and the need to leave enough room for selecting good start times for the other activities that have not yet been scheduled.

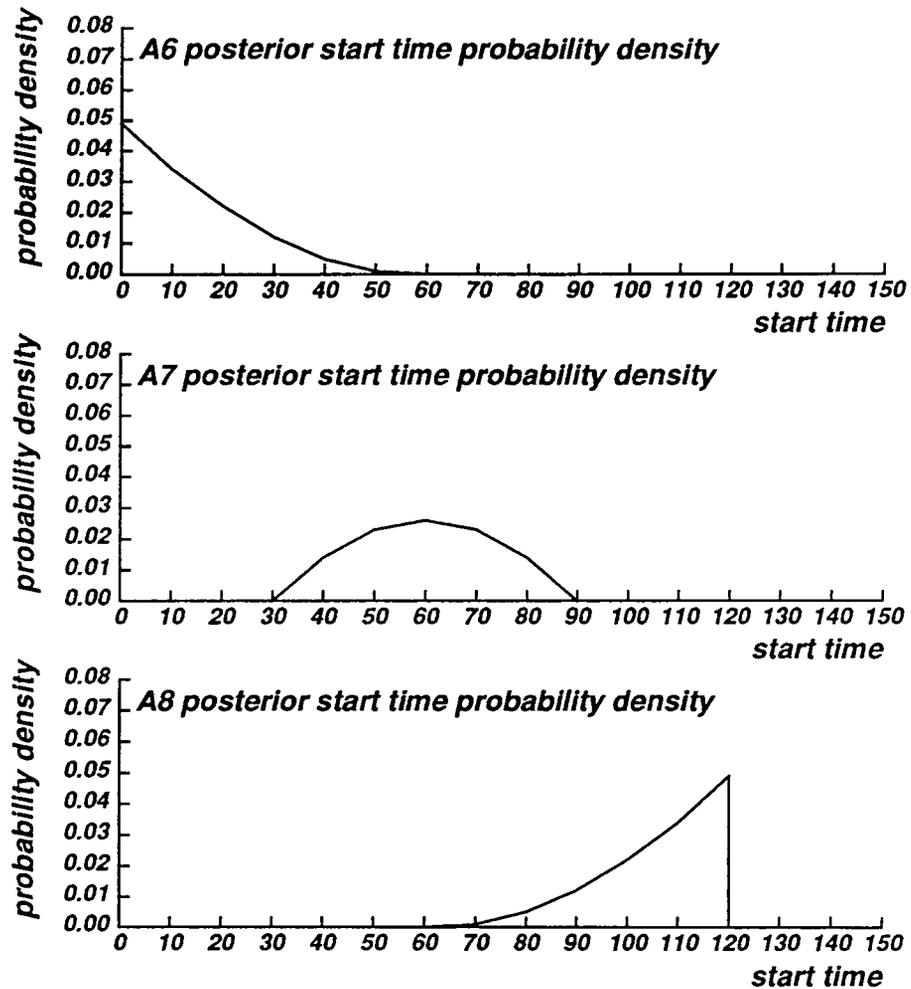


Figure 4-6: A posteriori start time densities for order₂

4.3. Propagation in an Acyclic TCG with Variable-duration Activities

In the case of variable-duration activities, one can compute for each activity a two-dimensional joint a posteriori probability density of the activity's start time and duration. This density, of the form $P(st_0=t \& du_0=d \& C_1 \& C_2 \& \dots \& C_m)$, represents the probability that $st_0=t$ and $du_0=d$ and that all the temporal relation constraints are satisfied given the activities' a priori start time and duration distributions. The integrals involved in the computation of these distributions are very similar to those of the previous subsection, except that we now have to account for the a priori duration distributions.

The equivalent to equations (2) and (3) are:

$$P(st_0 = t \& du_0 = d \& C_1 \& C_2 \& \dots \& C_m) = \sigma(st_0 = t) \times \delta_0(du_0 = d) \times P(C_1 \& C_2 \& \dots \& C_m | st_0 = t \& du_0 = d) \quad (9)$$

with:

$$P(C_1 \& C_2 \& \dots \& C_m | st_0 = t \& du_0 = d) = \prod_{i=1}^{p_0} P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \quad (10)$$

Paragraphs 4.3.1 and 4.3.2 develop the computation of $P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d)$ in terms of $P(st_i^0 = \tau \& du_i^0 = \delta \& S_i^0) = \sigma_i^0(st_i^0 = \tau) \delta_i^0(du_i^0 = \delta) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta)$ in the case where C_i^0 is respectively of the form "I₀ MEETS I_i⁰", and "I₀ BEFORE I_i⁰". The treatment of the set of all thirteen of Allen's temporal relation constraints can be found in appendix1. Exactly like in the previous subsection, one can use these equations in a recursive fashion to express the a posteriori start time and duration densities in terms of the a priori ones.

4.3.1. C_i⁰: I₀ MEETS I_i⁰

As before, the constraint "I₀ MEETS I_i⁰" requires (Figure 4-4) that I₀'s end time be equal to I_i⁰'s start time, i.e. it requires that $et_0 = st_0 + du_0 = st_i^0$. Hence, assuming that $st_0 = t$ and $du_0 = d$, the probability that C_i⁰ and the constraints in S_i⁰ are satisfied is equal to the probability that $st_i^0 = t + d$ and that the constraints in S_i⁰ are satisfied:

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) &= P(st_i^0 = t + d \& S_i^0) \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) \sigma_i^0(st_i^0 = t + d) P(S_i^0 | st_i^0 = t + d \& du_i^0 = \delta) d\delta \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_0^\infty \beta^1(\tau = t + d) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \end{aligned} \quad (11)$$

The first equality is the most natural one. We will however use equation (11) in the next subsection, when allowing for cycles in the TCG.

4.3.2. C_i⁰: I₀ BEFORE I_i⁰

In the same fashion, if I₀ has to be BEFORE I_i⁰, one has:

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{t+d}^\infty \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \end{aligned} \quad (12)$$

4.4. Relaxing the Acyclicity Assumption

We now turn our attention to the case where there may be cycles in the TCG. Equation (9) still holds but the computation of $P(C_1 \& C_2 \& \dots \& C_m | st_0 = t \& du_0 = d)$ becomes more complex.

$P(C_1 \& C_2 \& \dots \& C_m | st_0 = t \& du_0 = d)$ is the probability that C₁, C₂, ..., C_m are satisfied when $st_0 = t$ and $du_0 = d$ given the activities' a priori start time and duration distributions. The set of

temporal relation constraints $\{C_1, C_2, \dots, C_m\}$ can be expressed as a set of linear equalities and inequalities (e.g. I_i BEFORE I_j is equivalent to $st_i + du_i < st_j$). This set of equalities and inequalities together with the conditions $st_0 = t$ and $du_0 = d$ defines a polyhedron in the $2(n-1)$ dimensional space generated by $st_1^*, st_2^*, \dots, st_{n-1}^*, du_1^*, du_2^*, \dots, du_{n-1}^*$ ¹¹. The volume contained in this polyhedron is the domain of admissible values for $st_1^*, st_2^*, \dots, st_{n-1}^*, du_1^*, du_2^*, \dots, du_{n-1}^*$ given the TCG and the conditions $st_0 = t$ and $du_0 = d$ (independently of the a priori start time and duration distributions). Therefore $P(C_1 \& C_2 \& \dots \& C_m | st_0 = t \& du_0 = d)$ can be obtained by integrating the multivariable probability density $\sigma_1^*(st_1^* = \tau_1) \sigma_2^*(st_2^* = \tau_2) \dots \sigma_{n-1}^*(st_{n-1}^* = \tau_{n-1}) \delta_1^*(du_1^* = \delta_1) \delta_2^*(du_2^* = \delta_2) \dots \delta_{n-1}^*(du_{n-1}^* = \delta_{n-1})$ over this volume. In this subsection we explain how to effectively build this multiple integral as an *iterated integral*.

Notice that, according to Fubini's theorem (see [Thomas 83] for instance), there are $[2(n-1)]!$ correct ways to express a $2(n-1)$ -tuple integral as an iterated integral (each corresponding to a permutation of the $2(n-1)$ integration variables). The algorithm that we present builds one of these $[2(n-1)]!$ iterated integrals. Although all the iterated forms are theoretically equivalent, some result in faster numerical evaluation than others¹². The algorithm that we describe gives one way to build these integrals. The integrals can then be rearranged in order to speed up their evaluations. We will not be concerned here with these implementation details.

Consider again the TCG associated to order₂ in the example of subsection 1.3 (Figure 4-7)¹³.

As we saw in equation (6):

$$P(C_6 \& C_7 | st_6 = t) = \int_{t+du_6}^{\infty} \sigma_7(st_7 = \tau_7) d\tau_7 \int_{\tau_7+du_7}^{\infty} \sigma_8(st_8 = \tau_8) d\tau_8 \quad (13)$$

Alternatively, we can start integrating on st_8 , which produces:

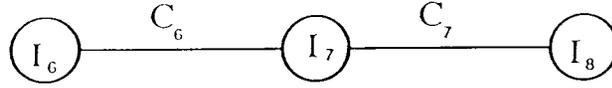
$$\begin{aligned} P(C_6 \& C_7 | st_6 = t) &= \int_{-\infty}^{+\infty} \sigma_8(st_8 = \tau_8) \alpha(\tau_8 - du_7 > t + du_6) d\tau_8 \int_{t+du_6}^{\tau_8 - du_7} \sigma_7(st_7 = \tau_7) d\tau_7 \\ &= \int_{t+du_6+du_7}^{\infty} \sigma_8(st_8 = \tau_8) d\tau_8 \int_{t+du_6}^{\tau_8 - du_7} \sigma_7(st_7 = \tau_7) d\tau_7 \end{aligned} \quad (14)$$

where $\alpha(\tau_8 - du_7 > t + du_6)$ simply expresses that the second integral's upper bound has to be greater than its lower bound (since we are integrating probability densities). Figure 4-7 represents the domain of integration of both form (13) and (14). They are obviously the same, which illustrates that both forms (both iterated integrals) are equivalent. Besides its illustration of Fubini's theorem, this example shows how to account for several constraints at the same time when determining a variable's domain of integration: in (14) the domain of integration of st_7 is determined by the two constraints C_6 and C_7 . In this example C_6 determines the lower bound of the integral and C_7 the upper bound. The simplicity of the formulas in the previous subsections was coming from the fact that it was possible to order the integration variables so as to account

¹¹Where $\{I_0, I_1^*, I_2^*, \dots, I_{n-1}^*\} = \{I_1, I_2, \dots, I_n\}$, I_0 being an arbitrary time interval of the set, as in the previous subsections. The "*" simply indicates that the time intervals have been reordered.

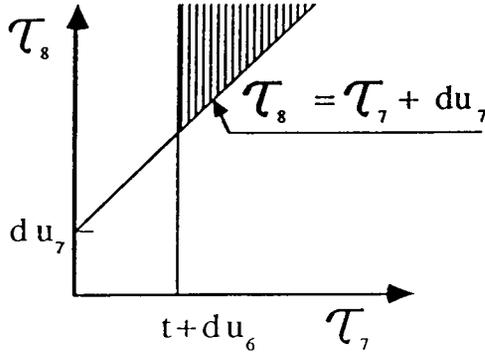
¹²Some iterated forms are also easier to solve analytically than others.

¹³One of the reasons for choosing this example is that the domain of integration can be visualized in 2-D.

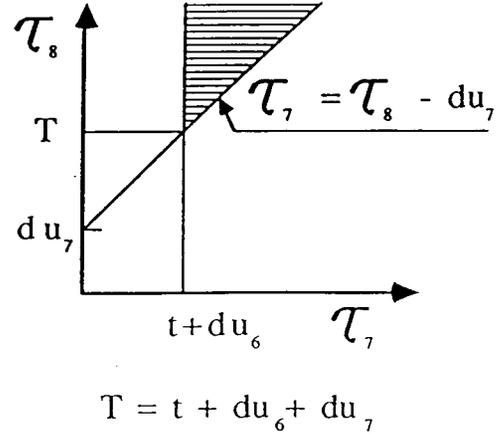


C_6 : I_6 BEFORE I_7

C_7 : I_7 BEFORE I_8



domain of integration in (13)



domain of integration in (14)

Figure 4-7: Illustration of Fubini's Theorem in a TCG with 3 Time Periods

for only one constraint at a time. In TCGs with cycles it is generally not possible to find such an ordering. The domain of integration of a variable is usually determined by several constraints, some affecting the lower-bound some the upper-bound. The actual lower-bound will therefore be given by the maximum of the lower-bounds produced by each constraint (i.e. the most restrictive one) and the actual upper-bound by the minimum of the upper-bounds produced by each constraint. Additionally one has to ensure that the lower-bound is smaller than the upper-bound (see α function in (14)) since we are integrating probability densities.

Order₁ (Figure 1-2) in subsection 1.3, is an example of a TCG with cycle where the integration bounds of some start times are obtained by taking the minimum or maximum of the bounds produced by several constraints. For instance:

$$\begin{aligned}
 P(C_1 \& C_2 \dots \& C_5 | st_1 = t) = \int_{t+du_1}^{\infty} \sigma_2(st_2 = \tau_2) d\tau_2 \int_{t+du_1}^{\infty} \sigma_4(st_4 = \tau_4) d\tau_4 \\
 \int_{\tau_2+du_2}^{\infty} \sigma_3(st_3 = \tau_3) d\tau_3 \int_{\text{Max}\{\tau_3+du_3, \tau_4+du_4\}}^{\infty} \sigma_5(st_5 = \tau_5) d\tau_5
 \end{aligned} \tag{15}$$

As suggested by the above example, the procedure for building iterated integrals to compute a

posteriori probability distributions in TCG with cycles is just a generalization of the formulas given in the previous subsections. The main differences come from the fact that it is not possible anymore to find an ordering of the integration variables that would allow for accounting for only one temporal relation constraint at a time. We just saw how to combine the lower-bounds and upper-bounds imposed by different temporal constraints. Before describing a general procedure to effectively build a posteriori probability integrals, we still have one detail to consider. Some of the formulas given in the previous subsection include β distributions (see also appendix 1). For instance $\beta^l(\tau=t+d)$ in (11) expresses that st_i^0 should be equal to $t+d$ in order for I_0 to meet I_1^0 . β distributions provide an easy way to formally handle all temporal relation constraints in the same fashion, i.e. with integrals over the duration and start time of each time interval. β distributions allow for expressing equalities involving integration variables. When several constraints involving β distributions affect the same time interval, one has to make sure that the values for the interval's start time (or duration) that they each require are compatible. This is accomplished by using the following rule (which can easily be verified using the definition of β distributions):

$$\int_{Domain} \beta^l[EQ_1, \dots, EQ_l] \beta^m[EQ_{l+1}, \dots, EQ_{l+m}] g(\xi_1, \dots, \xi_n) d\xi_1 \dots d\xi_n \\ = \int_{Domain} \beta^{l+m}[EQ_1, \dots, EQ_{l+m}] g(\xi_1, \dots, \xi_n) d\xi_1 \dots d\xi_n \quad (16)$$

This is illustrated by the example below.

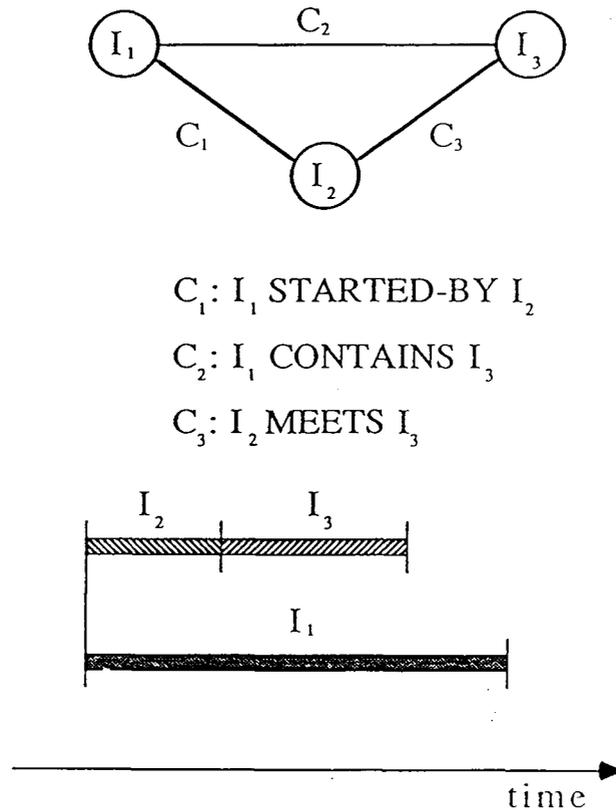


Figure 4-8: A TCG with 3 Time Periods

The TCG represented in Figure 4-8 involves 3 time intervals (with variable durations), namely I_1 , I_2 , and I_3 . The temporal relation constraints are:

- C_1 : I_1 STARTED-BY I_2 ,
- C_2 : I_1 CONTAINS I_3 , and
- C_3 : I_2 MEETS I_3 .

Using equations (39), (41), (35) (see appendix), and (16) one can write:

$$P(C_1 \& C_2 \& C_3 | st_1=t \& du_1=d) = \int_0^d \delta_3(du_3=\varepsilon_3) d\varepsilon_3 \int_t^{t+d-\varepsilon_3} \sigma_3(st_3=\tau_3) d\tau_3 \\ \int_{\text{Max}\{0,0\}}^{\text{Min}\{\infty,d\}} \delta_2(du_2=\varepsilon_2) d\varepsilon_2 \int_{\text{Max}\{-\infty,-\infty\}}^{\text{Min}\{+\infty,+\infty\}} \beta^2(\tau_2=t, \tau_2=\tau_3-\varepsilon_2) \sigma_2(st_2=\tau_2) d\tau_2$$

This formula can be simplified using the definition of β distributions:

$$P(C_1 \& C_2 \& C_3 | st_1=t \& du_1=d) \\ = \int_0^d \delta_3(du_3=\varepsilon_3) d\varepsilon_3 \int_t^{t+d-\varepsilon_3} \sigma_3(st_3=\tau_3) \alpha(0 < \tau_3 - t < d) \delta_2(du_2=\tau_3 - t) \sigma_2(st_2=t) d\tau_3 \\ = \sigma_2(st_2=t) \int_0^d \delta_3(du_3=\varepsilon_3) d\varepsilon_3 \int_t^{t+d-\varepsilon_3} \sigma_3(st_3=\tau_3) \delta_2(du_2=\tau_3 - t) d\tau_3$$

To conclude this subsection, Figure 4-9 gives a description of BUILD-A-POSTERIORI-PROBABILITY-EXPRESSION, a general procedure to effectively express $P(C_1, \dots, C_m | st_0=t \& du_0=d)$ as an iterated integral. The body of the procedure makes use of a couple of simple functions, of which we only give an informal description:

- **adjacent**(I , TCG): returns a list containing the time intervals adjacent to I in the TCG , I being itself a time interval.
- **pop**($list$): removes the first element from $list$ and returns it.
- **index**(I): returns the index (i.e. subscript) of I , where I is a time interval (e.g. **index**(I_2) returns 2).
- **intersection**($list_1, list_2$): returns a list containing the elements of $list_1$ that are also in $list_2$ (the order of the elements in the result list is arbitrary).
- **union**($list_1, list_2$): returns a list containing any element that is either in $list_1$ or $list_2$ (or in both). An element appearing in both $list_1$ and $list_2$ is returned only once.
- **list-difference**($list_1, list_2$): returns a list with the elements of $list_1$ that are not in $list_2$.
- **start-time-upper-bound-expression**(I , $list$): I is a time interval, and $list$ is a list of time intervals adjacent to I in the TCG . The function returns the start time upper-bound expression resulting from the temporal relation constraints between I and the time intervals in $list$. As explained earlier in this subsection, this is expressed as the minimum of the upper-bound produced by each constraint.
- **start-time-lower-bound-expression**(I , $list$): same as above. The lower-bound is expressed as the maximum of the lower-bound produced by each constraint.

- **duration-upper-bound-expression**($I, list$): same as above for the duration (minimum).
- **duration-lower-bound-expression**($I, list$): same as above for the duration (maximum).
- **beta-expression**($I, list$): combines the eventual β distributions resulting from the temporal constraints between I and the time intervals in $list$, as explained earlier in this subsection. If there are no β distributions the function simply returns the empty expression.
- **append**($expression_1, expression_2$): appends the two expressions together.

We use ε and **nil** to respectively denote the empty expression and the empty list.

procedure BUILD-A-POSTERIORI-PROBABILITY-EXPRESSION (I_0, TCG)

```

INTERVALS-TO-BE-PROCESSED  $\leftarrow$  adjacent( $I_0, TCG$ ) ;
PARTIAL-EXPRESSION  $\leftarrow$   $\varepsilon$  ;
MARKED-INTERVALS  $\leftarrow$  { $I_0$ } ;
      *a list containing  $I_0$ *
while INTERVALS-TO-BE-PROCESSED  $\neq$  nil
   $I \leftarrow$  pop(INTERVALS-TO-BE-PROCESSED) ;
   $i \leftarrow$  index( $I$ ) ;
  RELATED-INTERVALS  $\leftarrow$  adjacent( $I, TCG$ ) ;
  INTERVALS-TO-ACCOUNT-FOR  $\leftarrow$ 
    intersection(MARKED-INTERVALS, RELATED-INTERVALS) ;
  SUB  $\leftarrow$  start-time-upper-bound-expression( $I, INTERVALS-TO-ACCOUNT-FOR$ ) ;
  SLB  $\leftarrow$  start-time-lower-bound-expression( $I, INTERVALS-TO-ACCOUNT-FOR$ ) ;
  DUB  $\leftarrow$  duration-upper-bound-expression( $I, INTERVALS-TO-ACCOUNT-FOR$ ) ;
  DLB  $\leftarrow$  duration-lower-bound-expression( $I, INTERVALS-TO-ACCOUNT-FOR$ ) ;
  BETA  $\leftarrow$  beta-expression( $I, INTERVALS-TO-ACCOUNT-FOR$ ) ;
  LOCAL-EXPR  $\leftarrow$   $\alpha(DUB > DLB) \int_{DLB}^{DUB} \delta_i(du_i = \varepsilon_i) \alpha(SUB > SLB) d\varepsilon_i \int_{SLB}^{SUB} BETA \sigma_i(st_i = \tau_i) d\tau_i$  ;
  PARTIAL-EXPRESSION  $\leftarrow$  append(PARTIAL-EXPRESSION, LOCAL-EXPR) ;
  MARKED-INTERVALS  $\leftarrow$  union(MARKED-INTERVALS, { $I$ } ) ;
  INTERVALS-TO-BE-PROCESSED  $\leftarrow$  union(INTERVALS-TO-BE-PROCESSED,
    list-difference(RELATED-INTERVALS, MARKED-INTERVALS)) ;
while-end ;
return PARTIAL-EXPRESSION ;

```

Figure 4-9: Procedure to express $P(C_1, \dots, C_m | st_0 = t \& du_0 = d)$ as an iterated integral

The procedure builds the iterated integral from left to right by successively visiting each time interval that is directly or indirectly related to I_0 . LOCAL-EXPR contains the integrals over the start time and duration of the time interval currently visited. This local expression is appended to the right of a current partial expression of the iterated integral, thereby resulting in a new partial expression. Intervals that have been visited (i.e. whose start time and duration a priori densities have already been integrated in PARTIAL-EXPRESSION) are marked. Integration bounds for a time

interval's start time and duration are determined by the temporal relation constraints between that time interval and the adjacent time intervals that have already been marked.

As illustrated in the previous examples, the expressions produced by this procedure can be simplified using the definitions of α functions and β distributions. The integration bounds can also be refined to account for the very domain over which the probability densities are strictly positive. Finally the order of integration can be rearranged to speed up evaluation. A time complexity analysis of the method and a discussion of available methods to evaluate the integrals are given in section 6.

Figure 4-10 illustrates the operation of the procedure in the construction of the iterated integral in (15). Notice that the α expressions have been omitted as they trivially evaluate to 1.

Figure 4-11 displays the a posteriori start time densities of the activities in order₁, assuming no prior resource reservations. The start time utility functions are the ones described in subsection 1.3, triangle shaped utility functions allowing for start times between 0 and 140 with a peak in 120. One should notice the difference with the propagation of the uniform start time utilities of order₂ (Figure 4-6). For instance in the case of A_5 , the a posteriori density was not totally pushed to the right. Instead the density peaks at 130, which is a compromise between the optimal start time (120) and the tendency of the other activities to push A_5 towards its latest start time (140) in order to have more freedom. A similar remark applies to the other four activities. It should also be noted that A_1 's a posteriori start time density between 40 and 50 and A_5 's between 90 and 100 are not zero, though very small, which unfortunately does not appear very clearly on the graphs.

initialization:

PARTIAL-EXPRESSION: ε
 MARKED-INTERVALS: $\{I_1\}$
 INTERVALS-TO-BE-PROCESSED: $\{I_2, I_4\}$

step₁:

I: I_2
 PARTIAL-EXPRESSION: $\int_{t+du_1}^{\infty} \sigma_2(st_2=\tau_2) d\tau_2$
 MARKED-INTERVALS: $\{I_1, I_2\}$
 INTERVALS-TO-BE-PROCESSED: $\{I_4, I_3\}$

step₂:

I: I_4
 PARTIAL-EXPRESSION: $\int_{t+du_1}^{\infty} \sigma_2(st_2=\tau_2) d\tau_2 \int_{t+du_1}^{\infty} \sigma_4(st_4=\tau_4) d\tau_4$
 MARKED-INTERVALS: $\{I_1, I_2, I_4\}$
 INTERVALS-TO-BE-PROCESSED: $\{I_3, I_5\}$

step₃:

I: I_3
 PARTIAL-EXPRESSION: $\int_{t+du_1}^{\infty} \sigma_2(st_2=\tau_2) d\tau_2 \int_{t+du_1}^{\infty} \sigma_4(st_4=\tau_4) d\tau_4 \int_{\tau_2+du_2}^{\infty} \sigma_3(st_3=\tau_3) d\tau_3$
 MARKED-INTERVALS: $\{I_1, I_2, I_3, I_4\}$
 INTERVALS-TO-BE-PROCESSED: $\{I_5\}$

step₄:

I: I_5
 PARTIAL-EXPRESSION: $\int_{t+du_1}^{\infty} \sigma_2(st_2=\tau_2) d\tau_2 \int_{t+du_1}^{\infty} \sigma_4(st_4=\tau_4) d\tau_4 \int_{\tau_2+du_2}^{\infty} \sigma_3(st_3=\tau_3) d\tau_3$
 $\int_{\text{Max}\{\tau_3+du_3, \tau_4+du_4\}}^{\infty} \sigma_5(st_5=\tau_5) d\tau_5$
 MARKED-INTERVALS: $\{I_1, I_2, I_3, I_4, I_5\}$
 INTERVALS-TO-BE-PROCESSED: **nil**

Figure 4-10: Main steps involved in the construction of (15).

Notice that the α expressions have been omitted as they trivially evaluate to 1.

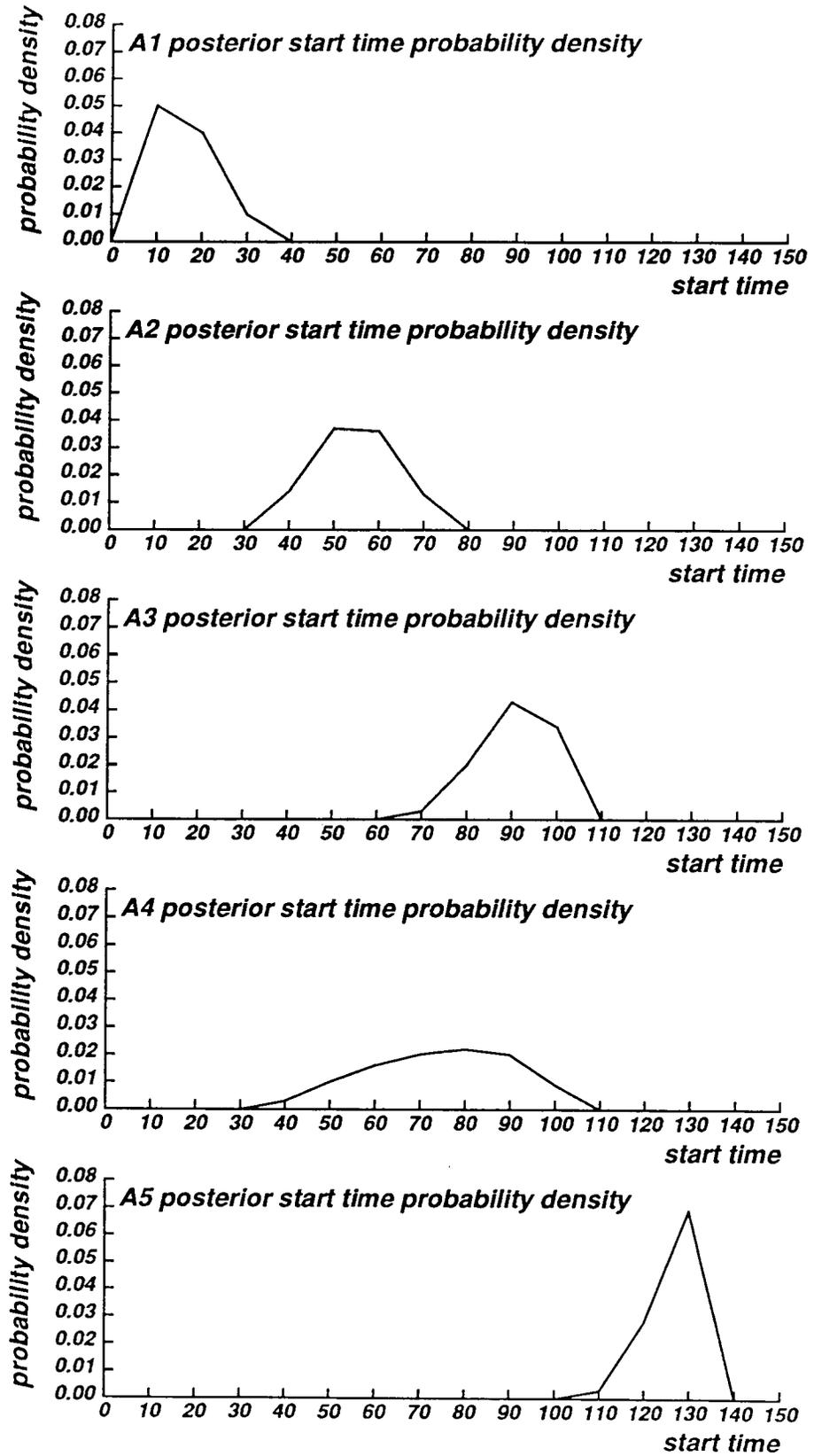


Figure 4-11: A posteriori start time densities for order₁

4.5. Propagation in a TCG with Explicit Disjunctions

In the case of explicit disjunctions in the TCG, one has to add the probabilities of all possible combinations of relations. Consider the example displayed in Figure 4-12. There are three time intervals I_1 , I_2 , and I_3 . The temporal relation constraints are:

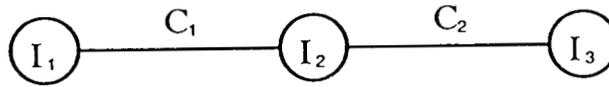
- C_1 : I_1 {BEFORE, AFTER} I_2 , and
- C_2 : I_2 {BEFORE, AFTER} I_3 .

Four combinations are possible:

1. I_1 BEFORE I_2 and I_2 BEFORE I_3 ,
2. I_1 BEFORE I_2 and I_2 AFTER I_3 ,
3. I_1 AFTER I_2 and I_2 BEFORE I_3 , and
4. I_1 AFTER I_2 and I_2 AFTER I_3 .

The a posteriori probability distributions are obtained by adding the probabilities of all four possibilities. For example:

$$\begin{aligned}
 &P(C_1 \& C_2 | st_1 = t) \\
 &= \int_{t+du_1}^{\infty} \sigma_2(st_2 = \tau_2) d\tau_2 \int_{\tau_2+du_2}^{\infty} \sigma_3(st_3 = \tau_3) d\tau_3 \\
 &+ \int_{t+du_1}^{\infty} \sigma_2(st_2 = \tau_2) d\tau_2 \int_{-\infty}^{\tau_2-du_3} \sigma_3(st_3 = \tau_3) d\tau_3 \\
 &+ \int_{-\infty}^{t-du_2} \sigma_2(st_2 = \tau_2) d\tau_2 \int_{\tau_2+du_2}^{\infty} \sigma_3(st_3 = \tau_3) d\tau_3 \\
 &+ \int_{-\infty}^{t-du_2} \sigma_2(st_2 = \tau_2) d\tau_2 \int_{-\infty}^{\tau_2-du_3} \sigma_3(st_3 = \tau_3) d\tau_3 \\
 &= \int_{t+du_1}^{\infty} \sigma_2(st_2 = \tau_2) [1 - \int_{\tau_2-du_3}^{\tau_2+du_2} \sigma_3(st_3 = \tau_3) d\tau_3] d\tau_2 \\
 &+ \int_{-\infty}^{t-du_2} \sigma_2(st_2 = \tau_2) [1 - \int_{\tau_2-du_3}^{\tau_2+du_2} \sigma_3(st_3 = \tau_3) d\tau_3] d\tau_2
 \end{aligned}$$



$$C_1: I_1 \{ \text{BEFORE, AFTER} \} I_2$$

$$C_2: I_2 \{ \text{BEFORE, AFTER} \} I_3$$

Figure 4-12: A TCG with explicit disjunctions

As the number of possible combinations grows exponentially with the number of time intervals, the computations are expected to quickly become intractable. Fortunately, in the factory scheduling domain, it has been our experience that such disjunctions are extremely infrequent.

4.6. Result Interpretation

It is interesting at this point to look back at the desiderata and requirements identified in subsection 3.2 and check if they are satisfied. From subsection 3.2, we already know that our a priori start time distributions have been built to satisfy **requirement1**. In order to properly perform consistency checking we still need to check **requirement2**, i.e. we need to make sure that no admissible value may receive a zero a posteriori probability. This just follows from probability theory. Moreover the method will give a zero a posteriori probability to any value forbidden by the TCG, given the a priori probability distributions. Hence the computation of the a posteriori probabilities is perfect with respect to **desideratum2**, as far as the interactions defined by the TCG are concerned. The remaining inconsistencies result from the difficulty to account for inter-order interactions between unscheduled activities.

Once the a posteriori start time and duration distributions have been computed, one has to distinguish between two possible situations:

1. If at least one of the a posteriori probability density is uniformly zero then the current CSP is unsatisfiable (inconsistent). The incremental scheduler should backtrack, if still possible.
2. Otherwise, after having been normalized, the a posteriori distributions can be combined to obtain the resource demand densities induced by the CSP, as we describe in the next section¹⁴. The normalization simply expresses that the total probability that each activity occurs is equal to one.

Because they account for the interactions defined by the TCG, a posteriori start time (and duration) distributions reflect intra-order interactions. They generalize the Operations Research notion of activity slack [Johnson 74]. Indeed a value with a high a posteriori probability will usually correspond to a high utility and will be likely to leave a lot of freedom for selecting high utility values for the other variables that have not been assigned a value yet. Therefore selection of start times (and durations) with high a posteriori probabilities is expected to result in good solutions to the CSP (**desideratum1**)¹⁵. An activity whose range of admissible start times (and durations) with high a posteriori probabilities is very wide is an activity with a lot of slack (with respect to the TCG). On the other hand, if the range of admissible values with high a posteriori probabilities is small, the activity has little slack. Equivalently, from a constraint satisfaction point of view, these a posteriori distributions can be seen locally as measures of value goodness and globally as measures of variable looseness.

However it is important to understand that, in general, the peaks of the a posteriori start time and duration distributions will not exactly coincide with the optimal activity start times and durations of the problem, nor will they even coincide with those of the problem obtained by omitting the resource capacity constraints. For instance, in the case of order₁, the optimal start times of A₁, A₂, A₃, A₄, and A₅ are respectively 30, 60, 90, 60, and 120 (and 30, 60, 90, 90, and 120 if one omits R₂'s capacity constraint). Obviously these optimal start times do not exactly coincide with the peaks of the distributions displayed in Figure 4-11. This is because at this stage

¹⁴This situation is not a guarantee that the current CSP is satisfiable since we have only performed partial consistency checking.

¹⁵Although one should still account for inter-order interactions, which is the topic of the next section.

we have not accounted precisely for the interactions induced by the capacity constraints. Our a priori distributions accounted only implicitly for the existence of these interactions by assuming non zero probabilities for values that were not locally optimal (see Figure 3-1). Iterating the propagation process as suggested in subsection 3.2, i.e. using resource demand densities to guess new a priori probabilities, should improve the quality of the a posteriori start time and duration distributions as measures of start time and duration goodness (**desideratum1**).

5. Resource Demand Densities

We complete the propagation process by combining the a posteriori start time (and duration) probabilities to estimate the amount of contention for each resource. This is performed in two steps:

1. For each activity A_k , we compute a set of *individual demand densities* D_{kij} . For each resource R_{kij} that an activity A_k can use, the demand density $D_{kij}(t)$ reflects the probability that A_k uses R_{kij} at time t to fulfill its resource requirement R_{ki} . This probability depends both on the probability that A_k is active at time t and the probability that A_k uses R_{kij} to fulfill its requirement R_{ki} . The probability that an activity is active at some time t is given by the probability that the activity's start time and duration are such that the activity does not start after t and does not start so early that it is already finished by t . In the case of a fixed-duration activity A_k , this is the probability that the activity starts some time between $t-du_k$ and t . A detailed treatment of the computation of individual demand densities is given in appendix2. We will also interpret $D_{kij}(t)$ as the reliance of A_k on the possession of R_{kij} at time t . Indeed activities with little slack and few good possible resources will have high individual demand densities concentrated over short time periods and a few resources, whereas activities with a lot of slack and several good resource alternatives will have smoother individual demand densities spread over long periods of time and several resources (see appendix 2 for details).
2. For each resource, activities' individual demand densities are combined to obtain the resource's *aggregate demand density*. This density gives the expected demand for the resource as a function of time. In the example described in subsection 1.3, the aggregate demand densities are given by:

- R_1 's aggregate demand density = $D_1(t)=D_{111}(t)+D_{211}(t)+D_{511}(t)$
- R_2 's aggregate demand density = $D_2(t)=D_{112}(t)+D_{312}(t)+D_{412}(t)+D_{712}(t)$
- R_3 's aggregate demand density = $D_3(t)=D_{613}(t)+D_{713}(t)+D_{813}(t)$

Notice that the aggregation process is performed regardless of the resources' capacities. As a matter of fact, a resource's aggregate demand density at some time t may get larger than its capacity. In general high contention for a resource will require prompt attention from the scheduler.

Figure 5-1 depicts the aggregate demand densities $D_1(t)$, $D_2(t)$, and $D_3(t)$ for the example in subsection 1.3. Clearly the contention between A_3 , A_4 and A_7 for R_2 , which was predicted in the introduction, has been identified by the propagation method. It corresponds to the peak of $D_2(t)$ centered around $t=100$. This peak reaches a density of 1.5, which is much larger than any of the other peaks. Figure 5-2 shows the individual contributions of A_3 , A_4 , and A_7 to the demand around the peak, namely $D_{312}(t)$, $D_{412}(t)$, and $D_{712}(t)$. An area of width 30 has been

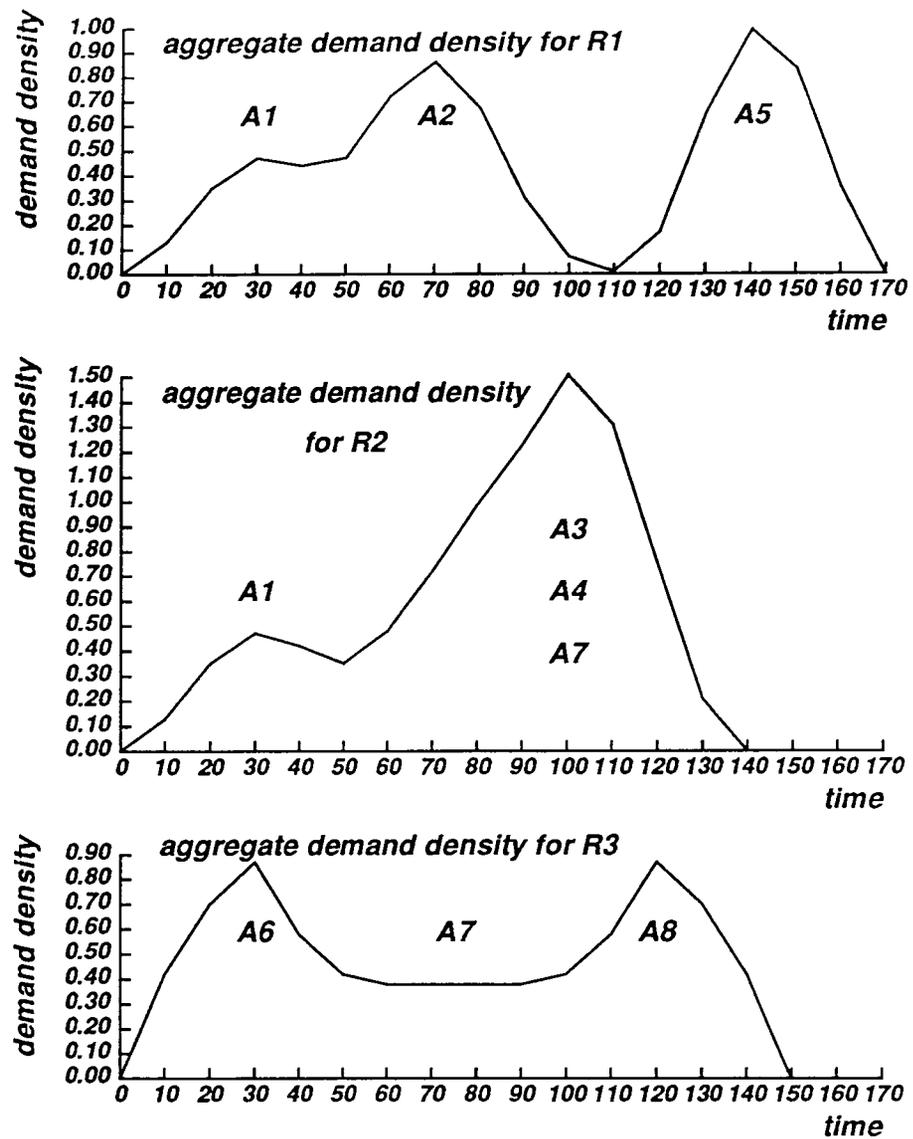


Figure 5-1: R_1 , R_2 , and R_3 's aggregate demand densities

delimited around the peak¹⁶. This is the area of high contention for R_2 . It clearly appears that within that zone, A_3 is the activity whose individual demand density contributes most to the demand for R_2 . Consequently A_3 is the activity that *relies* the most on the possession of R_2 within the area of high contention. An incremental scheduler can accordingly decide to first

¹⁶There is no particular reason for choosing 30 except that it seems to be a characteristic duration for this problem, since all the activities have a duration of 30. The same results would hold if we were considering slightly smaller or larger intervals of contention around the peak.

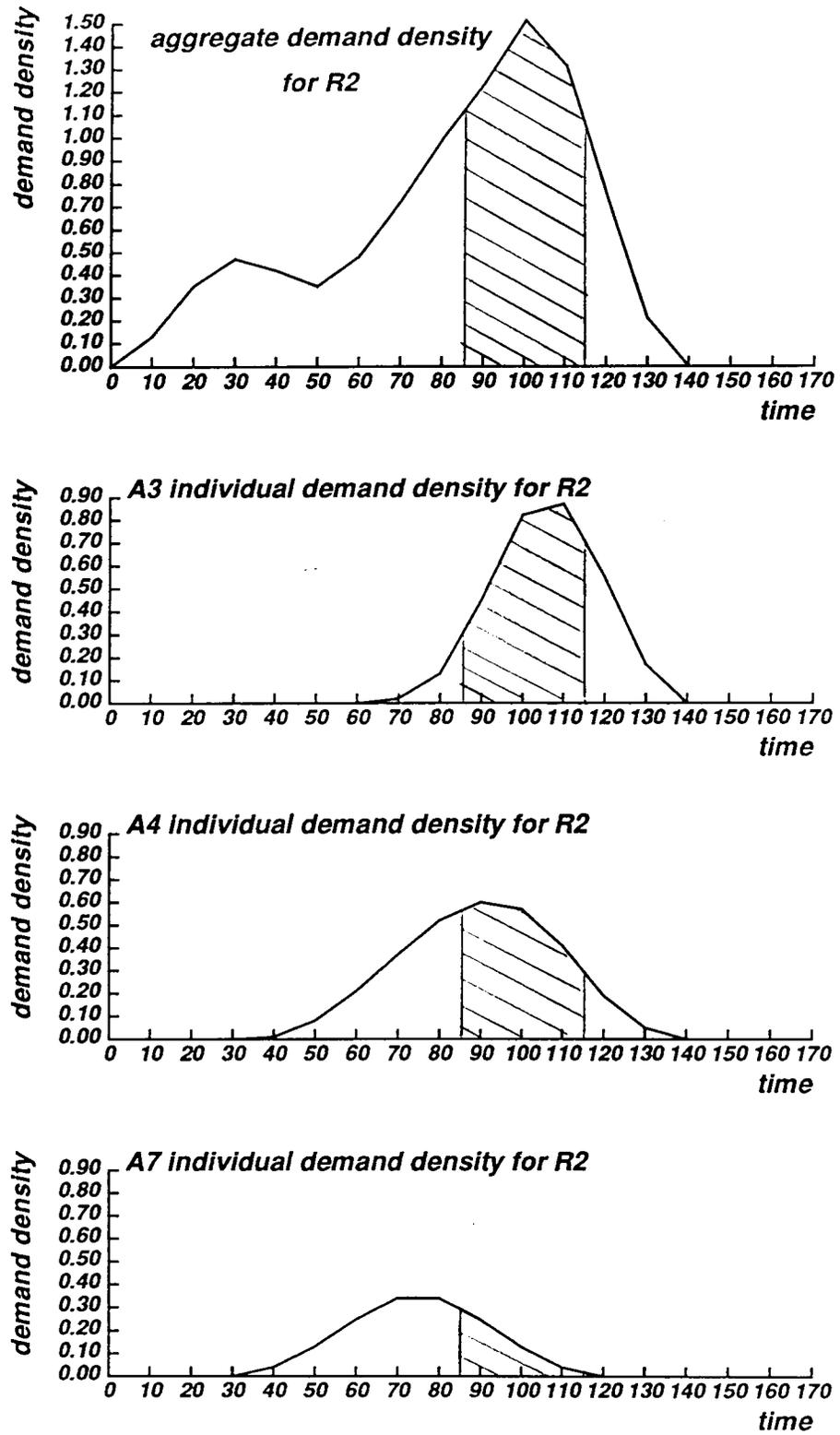


Figure 5-2: Contributions of A_3 , A_4 , and A_7 to R_2 's aggregate demand density focus its attention on the scheduling of A_3 .

6. Discussion

6.1. Time Complexity

As demonstrated in section 4, in a TCG with no explicit disjunctive constraints, $P(st_k = t \& du_k = d \& C_1 \& C_2 \& \dots \& C_m)$ can be expressed at worst with a $2(n-1)$ -tuple integral of a priori start-time and duration probability densities, where n is the number of activities to schedule. The construction of these integrals can be performed in polynomial time (see the procedure in Figure 4-9). On the other hand, evaluation of multiple integrals using classical integration techniques requires exponential time. In the worst case computation of the n a posteriori distributions requires $O(nK^{2n})$ integrand evaluations, where K is a constant that depends on the integration method. In the case of fixed-duration activities this complexity is still $O(nK^n)$. This exponential worst-case time complexity is actually a very pessimistic one. In manufacturing environments activities are grouped in orders. Only activities within the same order have temporal relation constraints between them. Therefore the largest multiple integrals that one has to evaluate correspond to the largest number of interconnected activities within an order (say $Max\ m_{order}$). This results in a worst case time complexity of $nK^{2Max\ m_{order}}$ in the case of n variable-duration activities, and a time complexity of $nK^{Max\ m_{order}}$ in the case of n fixed-duration activities. This also means that, for a given set of order types (i.e. $Max\ m_{order}$ is fixed), the worst-case time complexity to compute the a posteriori probability distributions is *linear in the number of orders to schedule*. The computation of the resource demand densities requires at most $O(n \times r)$ steps, for n activities and r resources. Hence for a fixed set of order types and a fixed set of resources, the asymptotic time complexity of the approach is linear in the number of orders to schedule.

In manufacturing environments, one may have to schedule up to several thousands of activities grouped in orders of up to 20 or 30 activities. Assuming that half of these activities are modeled as variable-duration activities, one may have multiple integrals of dimension up to 60. Numeric evaluation of such integrals is usually performed using *Monte Carlo* techniques [Stroud 71]. [Lepage 78] describes an adaptive Monte Carlo method for evaluating multidimensional integrals whose *asymptotic time and space complexities are linear in the integral's dimension*.

Alternatively one may try to reduce the size of the integrals via the use of a *hierarchical scheduler*.

6.2. Expressiveness of the Model

The preference propagation techniques that we have presented allow for all thirteen of Allen's temporal relation constraints as well as for disjunctions of such constraints. Additionally quantitative temporal relation constraints such as "*Activity_B* should start at least 5 minutes after *activity_A*" can be represented using dummy activities. For instance, one can introduce a dummy *activity_C* with duration of 5 minutes and the two constraints "*activity_A* MEETS *activity_C*" and "*activity_C* BEFORE *activity_B*". Using duration preferences one can express even more complex quantitative temporal relation constraints such as "*Activity_B* should start as soon as possible within 5 minutes after *activity_A*".

Our model accounts for three types of local preferential constraints: start time, duration, and resource preferential constraints. End time preferential constraints can be expressed using

dummy activities. For instance an end time preferential constraint on an *activity_A* can be expressed as a start time preferential constraint on a dummy *activity_B* MET-BY *activity_A*. Our framework also seems to allow for the representation of the most common global organizational constraints [Baker 74]. For instance, minimizing mean (weighted) order tardiness can be expressed with the help of end time (hence start time) constraints on the last activities of each order. Minimization of mean (weighted) order flowtime can be represented with aggregate activities, each containing all the activities in an order and a preferential constraint on the duration of each aggregate activity.

6.3. Possible Improvements

The preference propagation technique presented in this paper has been implemented on a Sun 3/60 running Knowledge Craft on top of Lucid Common Lisp for TCGs with fixed-duration activities interconnected by BEFORE/AFTER relations and that may contain cycles. An incremental scheduler has also been built that uses the preference propagation module to focus its attention. Preliminary experimentation with the system suggests several possible improvements.

6.3.1. Iterative and Hierarchical Preference Propagation

As already mentioned in subsection 3.2, we are currently investigating alternative ways to compute a priori probability distributions. In particular we are considering both iterative and hierarchical variations of the preference propagation scheme presented in this paper. In an iterative approach one can use the resource demand densities obtained by the previous iteration to estimate the probability that a given resource will be available for an activity at some time t . These probabilities can then be combined to obtain more accurate start time, duration, and resource a priori probability distributions for a new propagation. For instance good start times for which good resources are likely to be unavailable would see their a priori probability being reduced. A hierarchical approach is similar except that the additional information is obtained from the results of the propagation at the upper level rather than from the previous iteration. Such techniques are expected to account more accurately for the resource requirement interactions of unscheduled activities.

6.3.2. Activity Criticality

In the introduction we have defined a critical activity as one whose good (overall) start times and resources are likely to become unavailable if one started scheduling other activities first. In this paper we have assumed that the most critical activity is the one that relied the most on the possession of the most contended resource (over the area of high contention for that resource). This measure of activity criticality is only concerned with the availability of good resources at good start times. Good start times may however become unavailable just because of operation precedence interactions (i.e. intra-order interactions), as reflected in the a posteriori start time/duration distributions.¹⁷ We are looking for ways to integrate the notion of start time/duration looseness identified in subsection 4.6 directly into our measure of activity

¹⁷In the approach that we have presented, intra-order interactions are accounted for indirectly via the individual and aggregate demand densities, since these densities are computed from the a posteriori start time/duration distributions.

criticality rather than only indirectly through measures of resource contention and activity resource reliance. Additionally, rather than simply accounting for activity reliance with respect to the most contended resource, we would like to develop a measure that accounts for the reliance of an activity on each of its possible resources and the contention on each of these resources (over the appropriate time intervals).

6.3.3. Value Goodness

All along we have assumed that value goodness was solely determined by the problem constraints, i.e. both the required and preferential constraints of the problem. A more sophisticated approach would consist in also accounting for the time available to come up with a schedule. If there is very little time available, one will be mainly concerned with finding an admissible schedule as soon as possible. Good values are therefore the ones that are the least likely to result in backtracking, i.e. the least constraining values identified in earlier work in constraint satisfaction with uniformly preferred values [Haralick 80]. Instead if more time is available, it may be worthwhile considering riskier values because they are likely to result in a better schedule. For instance, if one machine is more accurate than all the other ones, one could try to schedule more activities on the most accurate machine. This may however result into some extra backtracking due to the higher contention for the accurate machine.

7. Summary and Concluding Remarks

Factory scheduling is subject to a wide variety of preferential constraints such as meeting due dates, reducing order flowtime, using accurate machines, etc. These local a priori preferences interact. For instance, meeting an order's due date may prevent the scheduler from selecting an accurate machine for an operation. Therefore selecting start times or resources based solely on such preferences is likely to result in poor schedules. *Preference propagation* strives for the construction of measures that reflect preference interactions. Such measures can then serve to guide the construction of good *overall* schedules rather than schedules that locally optimize a subset of a priori preferences.

Our approach to preference propagation is inspired by two CSP look-ahead techniques known as variable ordering and value ordering [Dechter 88]. Both theoretical and empirical studies [Haralick 80, Freuder 82, Nudel 83, Purdom 83, Stone 86] indicate that these techniques can significantly reduce the amount of search for a solution. Earlier work had only focused on applying these techniques to CSPs where variables have finite sets of equally preferred values. Our approach to preference propagation extends these techniques to CSPs where variables have infinite bounded sets of possible values with non-uniform preferences. The results of the propagation are formulated as a set of texture measures. In this paper we have identified the following texture measures: start time/duration goodness and looseness, resource contention, and activity resource reliance.

From an Operations Research point of view our preference propagation technique combines advantages of both order-based and resource-based scheduling by accounting for both intra-order and inter-order interactions [Smith 85].

We perform preference propagation within a probabilistic framework. A probability is associated to each variable's possible value that dynamically reflects the likelihood that the value results in a good schedule overall. We have identified requirements and desiderata to guide the

construction of such probabilities. These requirements and desiderata have been motivated by a double objective:

1. We want to be able to detect unsatisfiable CSPs as soon as possible (quick pruning), and
2. we want to use the propagation results to help focus the scheduler's attention on the most critical decision points and the most promising decisions at these points (opportunistic scheduling).

We have argued that the approach presented in this paper fulfills these requirements and desiderata.

We have described an algorithm to perform preference propagation in T/CCGs. The algorithm deals with all thirteen of Allen's temporal relation constraints and allows for cycles in the corresponding TCG. The algorithm also allows for activity start time, duration, and resource preferences and accounts for earlier resource reservations if any. We have shown that the results of the propagation across the temporal constraints can be combined to estimate resource contention and activity resource reliance. We have also analyzed the computational requirements of our approach.

The importance of this research lies in its attempt to give a more formal characterization of the problem space, in which we carry the search for a schedule. Given the underlying uncertainty of any search problem, a probabilistic characterization is a very attractive one. In this paper, we have presented a model that uses Bayesian probabilities to account for preference interactions in T/CCGs. The problem space is finally characterized by a set of textures that are used to guide the search process.

Appendix1: A Posteriori Start Time and Duration Distributions

In this appendix we summarize the essential formulas developed in subsections 4.2 and 4.3 and complete them to allow for all 13 of Allen's temporal relation constraints. The notations are the ones defined in subsection 4.1.

1. Acyclic TCG with fixed-duration activities

We found in subsection 4.2 that:

$$P(st_0 = t \& C_1 \& C_2 \& \dots \& C_m) = \sigma_0(st_0 = t) \times P(C_1 \& C_2 \& \dots \& C_m | st_0 = t) \quad (17)$$

with:

$$P(C_1 \& C_2 \& \dots \& C_m | st_0 = t) = \prod_{i=1}^{P_0} P(C_i^0 \& S_i^0 | st_0 = t) \quad (18)$$

C_i^0 may be any of Allen's thirteen temporal relation constraints:

1.1. C_i^0 : I_0 MEETS I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= P(st_i^0 = t + du_0 \& S_i^0) \\ &= \sigma_i^0(st_i^0 = t + du_0) P(S_i^0 | st_i^0 = t + du_0) \end{aligned} \quad (19)$$

1.2. C_i^0 : I_0 MET-BY I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= P(st_i^0 = t - du_i^0 \& S_i^0) \\ &= \sigma_i^0(st_i^0 = t - du_i^0) P(S_i^0 | st_i^0 = t - du_i^0) \end{aligned} \quad (20)$$

1.3. C_i^0 : I_0 BEFORE I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \int_{t+du_0}^{\infty} P(st_i^0 = \tau \& S_i^0) d\tau \\ &= \int_{t+du_0}^{\infty} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau) d\tau \end{aligned} \quad (21)$$

1.4. C_i^0 : I_0 AFTER I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \int_{-\infty}^{t-du_i^0} P(st_i^0 = \tau \& S_i^0) d\tau \\ &= \int_{-\infty}^{t-du_i^0} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau) d\tau \end{aligned} \quad (22)$$

1.5. C_i^0 : I_0 DURING I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \alpha(du_0 < du_i^0) \int_{t+du_0-du_i^0}^t P(st_i^0 = \tau \& S_i^0) d\tau \\ &= \alpha(du_0 < du_i^0) \int_{t+du_0-du_i^0}^t \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau) d\tau \end{aligned} \quad (23)$$

1.6. C_i^0 : I_0 CONTAINS I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \alpha(du_0 > du_i^0) \int_t^{t + du_0 - du_i^0} P(st_i^0 = \tau \& S_i^0) d\tau \\ &= \alpha(du_0 > du_i^0) \int_t^{t + du_0 - du_i^0} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau) d\tau \end{aligned} \quad (24)$$

1.7. C_i^0 : I_0 STARTS I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \alpha(du_0 < du_i^0) P(st_i^0 = t \& S_i^0) \\ &= \alpha(du_0 < du_i^0) \sigma_i^0(st_i^0 = t) P(S_i^0 | st_i^0 = t) \end{aligned} \quad (25)$$

1.8. C_i^0 : I_0 STARTED-BY I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \alpha(du_0 > du_i^0) P(st_i^0 = t \& S_i^0) \\ &= \alpha(du_0 > du_i^0) \sigma_i^0(st_i^0 = t) P(S_i^0 | st_i^0 = t) \end{aligned} \quad (26)$$

1.9. C_i^0 : I_0 FINISHES I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \alpha(du_0 < du_i^0) P(st_i^0 = t + du_0 - du_i^0 \& S_i^0) \\ &= \alpha(du_0 < du_i^0) \sigma_i^0(st_i^0 = t + du_0 - du_i^0) P(S_i^0 | st_i^0 = t + du_0 - du_i^0) \end{aligned} \quad (27)$$

1.10. C_i^0 : I_0 FINISHED-BY I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \alpha(du_0 > du_i^0) P(st_i^0 = t + du_0 - du_i^0 \& S_i^0) \\ &= \alpha(du_0 > du_i^0) \sigma_i^0(st_i^0 = t + du_0 - du_i^0) P(S_i^0 | st_i^0 = t + du_0 - du_i^0) \end{aligned} \quad (28)$$

1.11. C_i^0 : I_0 OVERLAPS I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \int_{\text{Max}\{t, t + du_0 - du_i^0\}}^{t + du_0} P(st_i^0 = \tau \& S_i^0) d\tau \\ &= \int_{\text{Max}\{t, t + du_0 - du_i^0\}}^{t + du_0} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau) d\tau \end{aligned} \quad (29)$$

1.12. C_i^0 : I_0 OVERLAPPED-BY I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \int_{t - du_i^0}^{\text{Min}\{t, t + du_0 - du_i^0\}} P(st_i^0 = \tau \& S_i^0) d\tau \\ &= \int_{t - du_i^0}^{\text{Min}\{t, t + du_0 - du_i^0\}} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau) d\tau \end{aligned} \quad (30)$$

1.13. C_i^0 : I_0 EQUALS I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t) &= \alpha(du_0 = du_i^0) P(st_i^0 = t \& S_i^0) \\ &= \alpha(du_0 = du_i^0) \sigma_i^0(st_i^0 = t) P(S_i^0 | st_i^0 = t) \end{aligned} \quad (31)$$

2. Acyclic TCG with variable-duration activities

We found in subsection 4.3 that:

$$\begin{aligned} P(st_0 = t \& du_0 = d \& C_1 \& C_2 \& \dots \& C_m) &= \sigma(st_0 = t) \times \delta_0(du_0 = d) \times \\ P(C_1 \& C_2 \& \dots \& C_m | st_0 = t \& du_0 = d) & \end{aligned} \quad (32)$$

with:

$$P(C_1 \& C_2 \& \dots \& C_m | st_0 = t \& du_0 = d) = \prod_{i=1}^{P_0} P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \quad (33)$$

C_i^0 may be any of Allen's thirteen temporal relation constraints:

2.1. C_i^0 : I_0 MEETS I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) &= P(st_i^0 = t + d \& S_i^0) \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) \sigma_i^0(st_i^0 = t + d) P(S_i^0 | st_i^0 = t + d \& du_i^0 = \delta) d\delta \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{+\infty} \beta^1(\tau = t + d) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \end{aligned} \quad (34)$$

The first equality is the most useful one. However equation (34) is useful for the treatment of TCGs with cycles (see subsection 4.4). The same remark applies to the other equations involving β distributions.

2.2. C_i^0 : I_0 MET-BY I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) &= \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) \sigma_i^0(st_i^0 = t - \delta) P(S_i^0 | st_i^0 = t - \delta \& du_i^0 = \delta) d\delta \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{+\infty} \beta^1(\tau = t - \delta) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \end{aligned} \quad (35)$$

2.3. C_i^0 : I_0 BEFORE I_i^0

$$\begin{aligned} P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) &= \\ &= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{t+d}^\infty \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \end{aligned} \quad (36)$$

2.4. C_i^0 : I_0 AFTER I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{t-\delta} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{37}$$

2.5. C_i^0 : I_0 DURING I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_d^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{t+d-\delta}^t \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{38}$$

2.6. C_i^0 : I_0 CONTAINS I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_0^d \delta_i^0(du_i^0 = \delta) d\delta \int_t^{t+d-\delta} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{39}$$

2.7. C_i^0 : I_0 STARTS I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_d^\infty \delta_i^0(du_i^0 = \delta) \sigma_i^0(st_i^0 = t) P(S_i^0 | st_i^0 = t \& du_i^0 = \delta) d\delta \\
&= \int_d^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{+\infty} \beta^1(\tau=t) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{40}$$

2.8. C_i^0 : I_0 STARTED-BY I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_0^d \delta_i^0(du_i^0 = \delta) \sigma_i^0(st_i^0 = t) P(S_i^0 | st_i^0 = t \& du_i^0 = \delta) d\delta \\
&= \int_0^d \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{+\infty} \beta^1(\tau=t) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{41}$$

2.9. C_i^0 : I_0 FINISHES I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_d^\infty \delta_i^0(du_i^0 = \delta) \sigma_i^0(st_i^0 = t+d-\delta) P(S_i^0 | st_i^0 = t+d-\delta \& du_i^0 = \delta) d\delta \\
&= \\
& \int_d^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{+\infty} \beta^1(\tau=t+d-\delta) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{42}$$

2.10. C_i^0 : I_0 FINISHED-BY I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_0^d \delta_i^0(du_i^0 = \delta) \sigma_i^0(st_i^0 = t + d - \delta) P(S_i^0 | st_i^0 = t + d - \delta \& du_i^0 = \delta) d\delta \\
&= \\
& \int_0^d \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{+\infty} \beta^1(\tau = t + d - \delta) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{43}$$

2.11. C_i^0 : I_0 OVERLAPS I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{\text{Max}\{t, t+d-\delta\}}^{t+d} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \\
&= \int_0^d \delta_i^0(du_i^0 = \delta) d\delta \int_{t+d-\delta}^{t+d} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \\
&+ \\
& \int_d^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_t^{t+d} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{44}$$

2.12. C_i^0 : I_0 OVERLAPPED-BY I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{t-\delta}^{\text{Min}\{t, t+d-\delta\}} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \\
&= \int_0^d \delta_i^0(du_i^0 = \delta) d\delta \int_{t-\delta}^t \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau \\
&+ \\
& \int_d^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{t-\delta}^{t+d-\delta} \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{45}$$

2.13. C_i^0 : I_0 EQUALS I_i^0

$$\begin{aligned}
& P(C_i^0 \& S_i^0 | st_0 = t \& du_0 = d) \\
&= \delta_i^0(du_i^0 = d) \sigma_i^0(st_i^0 = t) P(S_i^0 | st_i^0 = t \& du_i^0 = d) \\
&= \int_0^\infty \delta_i^0(du_i^0 = \delta) d\delta \int_{-\infty}^{+\infty} \beta^2(\delta = d, \tau = t) \sigma_i^0(st_i^0 = \tau) P(S_i^0 | st_i^0 = \tau \& du_i^0 = \delta) d\tau
\end{aligned} \tag{46}$$

Appendix2: Activity Individual Demand Densities

1. Notations

In this appendix we assume that a posteriori start time densities have already been computed as described in section 4. We assume that none of these densities is uniformly zero, otherwise this would indicate unsatisfiability of the current CSP and the incremental scheduler would have to backtrack. As already mentioned earlier, a posteriori start time densities can then be normalized to express the fact that each activity will occur once (i.e. each activity will start exactly once). These normalized a posteriori densities will be denoted:

- fixed-duration activities: $P_N(st_i=t \& C_1 \& \dots \& C_m)$
- variable-duration activities: $P_N(st_i=t \& du_i=d \& C_1 \& \dots \& C_m)$

$\rho_{R_{ki}}(R_{kij})$ will denote the a priori probability that A_k uses R_{kij} to fulfill its resource requirement R_{ki} . $D_{kij}(t)$ will denote A_k 's individual demand for R_{kij} as a function of t . This is the probability that A_k uses R_{kij} at time t to fulfill its resource requirement R_{ki} . The computations are performed assuming an incremental scheduler whose earlier resource reservations are non-preemptible. Therefore the demand density has to be reshaped so that it does not overlap with earlier reservations. We propose a method for doing so, which involves two steps.

Finally we will be using the predicate $AVAIL(R_{kij}, t, t+du_k)$ which returns **true** if and only if resource R_{kij} is available at all time between t and $t+du_k$. This is a precondition for scheduling activity A_k to start at time t , if A_k is to use resource R_{kij} .

2. Resource Demand Densities Produced by Fixed-Duration Activities

The probability that activity A_k uses R_{kij} at time t to fulfill its resource requirement R_{ki} is given by the a priori probability that A_k uses R_{kij} to fulfill R_{ki} multiplied by the conditional probability that A_k is active at time t given that it uses R_{kij} to fulfill R_{ki} . It turns out that this latter conditional probability may be uniformly zero for some resources R_{kij} due to earlier reservations. This can be accounted for by refining the a priori probabilities $\rho_{R_{ki}}(R_{kij})$. $D_{kij}(t)$ is therefore computed in two steps:

1. In the first step we compute:

$$D_{kij}^{step1}(t) = \rho_{R_{ki}}(R_{kij}) \int_{t-du_k}^t P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) d\tau$$

where $P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij})$ is the probability that $st_k=\tau$ and that the temporal relation constraints C_1, \dots, C_m are satisfied given the activities' a priori start time distributions and given that R_{kij} is the resource used to fulfill requirement R_{ki} . This probability can be approximated by computing I_k 's a posteriori start time distribution starting from an a priori start time distribution that accounts for R_{kij} 's reservations. We do so by replacing $\sigma_k(st_k=t)$ with $\sigma_k(st_k=t) \times \alpha[AVAIL(R_{kij}, t, t+du_k)]$ in the computation of $P(st_k=t \& C_1 \& \dots \& C_m)$. In other words:

$$P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) = \kappa P(st_k=\tau \& C_1 \& \dots \& C_m) \alpha[AVAIL(R_{kij}, \tau, \tau+du_k)]$$

where κ is a normalization factor¹⁸.

2. In the second step the a priori probabilities $\rho_{R_{ki}}(R_{kij})$ are refined to account for the resource reservations. The refined probabilities are denoted $\rho_{R_{ki}}^{\text{step2}}(R_{kij})$. Indeed, due to earlier reservations, some resources R_{kij} have a posteriori probabilities $P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij})$ that are uniformly zero. A_k 's individual demand for these resources is therefore uniformly zero as well. Hence one can use the new probabilities:

$$\rho_{R_{ki}}^{\text{step2}}(R_{kij}) = \begin{cases} 0 & \text{if } P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) \text{ is uniformly 0} \\ \kappa_{ki} \rho_{R_{ki}}(R_{kij}) & \text{otherwise} \end{cases}$$

where κ_{ki} is a normalization factor. Notice that, for each R_{ki} , because of the consistency checking performed after the computation of the a posteriori start time distributions, we are guaranteed at this point to have at least one resource R_{kij} such that $P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij})$ is not uniformly zero. One can then compute:

$$D_{kij}^{\text{step2}}(t) = \rho_{R_{ki}}^{\text{step2}}(R_{kij}) \int_{t-du_k}^t P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) d\tau$$

In practice it is not necessary to compute $D_{kij}^{\text{step1}}(t)$: one can just compute $P_N(st_k=\tau \& C_1 \& \dots \& C_m | R_{ki}=R_{kij})$ and check if it is uniformly zero or not.

Finally, notice that the total demand is given by:

$$\begin{aligned} & \int_{-\infty}^{+\infty} D_{kij}^{\text{step2}}(\tau) d\tau \\ &= \rho_{R_{ki}}^{\text{step2}}(R_{kij}) \int_{-\infty}^{+\infty} d\tau \int_{\tau-du_k}^{\tau} P_N(st_k=\xi \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) d\xi \\ &= \rho_{R_{ki}}^{\text{step2}}(R_{kij}) \int_{-\infty}^{+\infty} P_N(st_k=\xi \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) d\xi \int_{\xi}^{\xi+du_k} d\tau \quad (\text{using Fubini}) \\ &= \rho_{R_{ki}}^{\text{step2}}(R_{kij}) \times du_k \quad (\text{since } P_N \text{ is normalized}) \end{aligned}$$

and hence for each R_{ki} required by A_k :

$$\sum_j \int_{-\infty}^{+\infty} D_{kij}^{\text{step2}}(\tau) d\tau = du_k$$

which simply expresses that an activity A_k 's total demand for a resource R_{ki} is equal to its duration du_k . This duration has simply been distributed over time and over several resources (R_{kij}) to account for the different possible schedules of the activity.

¹⁸Again this normalization simply expresses that the activity will occur exactly once.

3. Resource Demand Densities Produced by Variable-Duration Activities

The computations in the case of variable-duration activities are very similar to the ones for fixed-duration activities:

1. In the first step one computes the distributions

$$P_N(st_k=\tau \& du_k=\varepsilon \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) = \kappa P(st_k=\tau \& du_k=\varepsilon \& C_1 \& \dots \& C_m) \\ \alpha[AVAIL(R_{kij}, \tau, \tau+\varepsilon)]$$

where κ is a normalization factor.

2. The probabilities $\rho_{R_{ki}}(R_{kij})$ are refined in the same way as for fixed-duration activities. One can then compute:

$$D_{kij}^{step2}(t) = \rho_{R_{ki}}^{step2}(R_{kij}) \int_0^\infty d\varepsilon \int_{t-\varepsilon}^t P_N(st_k=\tau \& du_k=\varepsilon \& C_1 \& \dots \& C_m | R_{ki}=R_{kij}) d\tau$$

Lastly, using Fubini's theorem, one can check that:

$$\begin{aligned} \int_{-\infty}^{+\infty} D_{kij}^{step2}(\tau_1) d\tau_1 &= \rho_{R_{ki}}^{step2}(R_{kij}) \int_{-\infty}^{+\infty} d\tau_1 \int_0^\infty d\varepsilon \int_{\tau_1-\varepsilon}^{\tau_1} P_N(st_k=\tau_2 \& du_k=\varepsilon \& C_1 \dots \& C_m | R_{ki}=R_{kij}) d\tau_2 \\ &= \rho_{R_{ki}}^{step2}(R_{kij}) \int_0^\infty d\varepsilon \int_{-\infty}^{+\infty} d\tau_2 \int_{\tau_2}^{\tau_2+\varepsilon} P_N(st_k=\tau_2 \& du_k=\varepsilon \& C_1 \dots \& C_m | R_{ki}=R_{kij}) d\tau_1 \\ &= \rho_{R_{ki}}^{step2}(R_{kij}) \int_0^\infty d\varepsilon \int_{-\infty}^{+\infty} \varepsilon P_N(st_k=\tau_2 \& du_k=\varepsilon \& C_1 \dots \& C_m | R_{ki}=R_{kij}) d\tau_2 \end{aligned}$$

Hence, for each R_{ki} required by A_k , A_k 's total demand is:

$$\sum_j \int_{-\infty}^{+\infty} D_{kij}^{step2}(\tau_1) d\tau_1 = \int_0^\infty d\varepsilon \int_{-\infty}^{+\infty} \varepsilon \left[\sum_j \rho_{R_{ki}}^{step2}(R_{kij}) P_N(st_k=\tau_2 \& du_k=\varepsilon \& C_1 \dots \& C_m | R_{ki}=R_{kij}) \right] d\tau_2$$

which is A_k 's expected duration given the joint start time and duration probability density $\sum_j \rho_{R_{ki}}^{step2}(R_{kij}) P_N(st_k=\tau_2 \& du_k=\varepsilon \& C_1 \dots \& C_m | R_{ki}=R_{kij})$.

Acknowledgement

We like to thank Nicola Muscettola and Katia Sycara for their comments on this work, and Joe Mattis for his help implementing the interface to the scheduler.

References

- [Allen 83] J.F.Allen.
Maintaining Knowledge about Temporal Intervals.
Communications of the ACM 26(11):832-843, 1983.
- [Allen 84] J.F.Allen.
Towards a General Theory of Action and Time.
Artificial Intelligence 23(2):123-154, 1984.
- [Baker 74] K.R. Baker.
Introduction to Sequencing and Scheduling.
Wiley, 1974.
- [Bell 84] Colin E. Bell and Austin Tate.
Using Temporal Constraints to Restrict Search in a Planner.
Technical Report AIAI-TR-5, Artificial Intelligence Applications Institute,
University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, United
Kingdom, 1984.
- [Dechter 88] Rina Dechter and Judea Pearl.
Network-Based Heuristics for Constraint Satisfaction Problems.
Artificial Intelligence 34(1):1-38, 1988.
- [Fox 83] M. Fox.
Constraint-Directed Search: A Case Study of Job-Shop Scheduling.
PhD thesis, Department of Computer Science, Carnegie-Mellon University,
1983.
- [Freuder 82] E.C. Freuder.
A Sufficient Condition for Backtrack-free Search.
Journal of the ACM 29(1):24-32, 1982.
- [Haralick 80] Robert M. Haralick and Gordon L. Elliott.
Increasing Tree Search Efficiency for Constraint Satisfaction Problems.
Artificial Intelligence 14(3):263-313, 1980.
- [Johnson 74] L.A. Johnson and D.C. Montgomery.
*Operations Research in Production Planning, Scheduling, and Inventory
Control*.
Wiley, 1974.
- [Lepage 78] G. Peter Lepage.
A New Algorithm for Adaptive Multidimensional Integration.
Journal of Computational Physics 27:192-203, 1978.
- [LePape 87] Claude Le Pape and Stephen F. Smith.
Management of Temporal Constraints for Factory Scheduling.
Technical Report, The Robotics Institute, Carnegie Mellon University,
Pittsburgh, PA 15213, 1987.
also appeared in Proc. Working Conference on Temporal Aspects in
Information Systems, Sponsored by AFCET and IFIP Technical
Committee TC8, North Holland Publishers, Paris, France, May 1987.

- [Mackworth 77] A.K. Mackworth.
Consistency in Networks of Relations.
Artificial Intelligence 8(1):99-118, 1977.
- [Muscettola 87] Nicola Muscettola, and Stephen Smith.
A Probabilistic Framework for Resource-Constrained Multi-Agent Planning.
In *Proceedings of the Tenth International Conference on Artificial Intelligence*, pages 1063-1066. 1987.
- [Nadel 86a] B.A. Nadel.
The General Consistent Labeling (or Constraint Satisfaction) Problem.
Technical Report DCS-TR-170, Department of Computer Science, Laboratory for Computer Research, Rutgers University, New Brunswick, NJ 08903, 1986.
- [Nadel 86b] B.A. Nadel.
Three Constraint Satisfaction Algorithms and Their Complexities: Search-Order Dependent and Effectively Instance-specific Results.
Technical Report DCS-TR-171, Department of Computer Science, Laboratory for Computer Research, Rutgers University, New Brunswick, NJ 08903, 1986.
- [Nadel 86c] B.A. Nadel.
Theory-based Search-order Selection for Constraint Satisfaction Problems.
Technical Report DCS-TR-183, Department of Computer Science, Laboratory for Computer Research, Rutgers University, New Brunswick, NJ 08903, 1986.
- [Nudel 83] Bernard Nudel.
Consistent-Labeling Problems and their Algorithms: Expected-Complexities and Theory-Based Heuristics.
Artificial Intelligence 21:135-178, 1983.
- [Ow 84] Peng Si Ow.
Heuristic Knowledge and Search for Scheduling.
Technical Report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, 1984.
(Ph.D. Thesis).
- [Purdom 83] Paul W. Purdom, Jr.
Search Rearrangement Backtracking and Polynomial Average Time.
Artificial Intelligence 21:117-133, 1983.
- [Rit 86] Jean-Francois Rit.
Propagating Temporal Constraints for Scheduling.
In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 383-388. 1986.
- [Smith 83] Stephen F. Smith.
Exploiting Temporal Knowledge to Organize Constraints.
Technical Report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, 1983.

- [Smith 85] Stephen F. Smith and Peng Si Ow.
The Use of Multiple Problem Decompositions in Time Constrained Planning Tasks.
In Proceedings of the Ninth International Conference on Artificial Intelligence, pages 1013-1015. 1985.
- [Smith 86] S. Smith, M. Fox, and P.S. Ow.
Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-Based Factory Scheduling Systems.
AI Magazine 7(4):45-61, Fall, 1986.
- [Stone 86] Harold S. Stone and Paolo Sipala.
The average complexity of depth-first search with backtracking and cutoff.
IBM Journal of Research and Development 30(3):242-258, 1986.
- [Stroud 71] A. H. Stroud.
Series in Automatic Computation: Approximate Calculations of Multiple Integrals.
Prentice Hall, 1971.
- [Thomas 83] Thomas and Finney.
Calculus and Analytic Geometry, Sixth Edition.
Addison Wesley, 1983.
- [Tsang 87] Edward P. K. Tsang.
The Consistent Labeling Problem in Temporal Reasoning.
In Proceedings of the Seventh National Conference on Artificial Intelligence, pages 251-255. 1987.
- [Valdes-Perez 87] Raul E. Valdes-Perez.
The Satisfiability of Temporal Constraint Networks.
In Proceedings of the Seventh National Conference on Artificial Intelligence, pages 256-260. 1987.
- [Vere 83] Stephen Vere.
Planning in Time: Windows and Durations for Activities and Goals.
IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5(3):246-267, 1983.
- [Vilain 86] Marc Vilain and Henry Kautz.
Constraint Propagation Algorithms for Temporal Reasoning.
In Proceedings of the Sixth National Conference on Artificial Intelligence, pages 377-382. 1986.

