CONTROLLING CONSTRAINT PROPAGATION

Anne Collinot
Universite Paris VI

Claude Le Pape
University Paris XI

Laboratoires de Marcoussis - Division DIN
Centre de Recherches de la CGE
Route de Nozay - 91460 Marcoussis - FRANCE

ABSTRACT

We have been engaged in work about the propagation of temporal constraints in the domain of job-shop scheduling. Many systems have been or are currently used to propagate temporal constraints. However, very few attempts have been made to build systems in which the amount of computational effort spent in constraint propagation is not fixed once for all. We do believe various pieces of knowledge can be used to dynamically adjust the amount of propagation. Consequently, we advocate the use of a flexible propagation system. We describe such a system and discuss control knowledge enabling dynamic adaptation of the amount of propagation to the needs of a problem solver. We also provide examples in the domain of job-shop scheduling.

I THE USEFULNESS OF CONTROLLING CONSTRAINT PROPAGATION

Constraint propagation is a deductive activity performed by a propagation system for a problem solver. It enables the problem solver to decompose a problem without neglecting interactions between subproblems, determine which subproblems are more constrained and focus attention accordingly. The propagation system also detects inconsistencies between three kinds of constraints:

- *Initial constraints* partially describe the problem to be solved.
- *Decisions* made by the problem solver refine these constraints and gradually reduce the search space.
- *Deviation constraints*, appearing in real-time, describe differences between the problem solver predictions and the actual course of events.

As soon as disjunctive constraints are considered, the problem of determining whether a given set of constraints is consistent is NP-hard. Therefore, two kinds of propagation systems can be distinguished. *Complete* systems guarantee consistency but are often useless for complexity reasons. *Polynomial* systems are (until *P=NP)* unable to immediately detect each contradiction. In general, a polynomial propagation system finds out a contradiction when the problem solver has already made decisions and refined constraints; a large amount of search is wasted since the problem solver explores spaces devoid of solutions. Similarly, all The consequences of a deviation are not immediately assessed. Therefore, the problem solver may be tardily warned of a need for plan revision and forced to make ill-considered reactive decisions.

Many constraint propagation systems have been or are currently used within problem solvers. Most of these perform a fixed amount of propagation - the trade-off between the anticipation of interactions and The amount of computational effort spent in constraint propagation is set once for all - and systems in which the amount of propagation can be modified are designed to reduce propagation with respect to a predetermined set of particular parameters (reference intervals in [Allen 1981], levels of precision in [Le Pape and Smith 1987]). *Ad-hoc* systems (e.g. [Le Pape and Smith 1987]), implemented as part of and in the formalism of particular problem solvers, efficiently deduce just enough information for their problem solver to work properly. However, additional analysis may sensibly improve the problem solver capabilities of pruning the search space and reacting to unanticipated events. On the other hand, *problem-solver independent* propagation systems (e.g. [Allen 1981]) are based on a theory and can be integrated into many problem solvers. Unfortunately, they may provide more or less information than needed. Indeed, the amount of propagation that enables a problem solver to be the most efficient varies from one problem to the other. It also varies dynamically during the search of a solution.

We do believe various pieces of knowledge can be used to dynamically adjust the amount of propagation. Consequently, we advocate the use of a flexible propagation system. Section II describes such a system designed for propagating temporal constraints. Section III is concerned with control knowledge allowing to adapt this system and provides examples in the domain of job-shop scheduling. Resulting benefits and remaining difficulties are discussed in section IV.

II A FLEXIBLE PROPAGATION SYSTEM

Our temporal propagation system consists of a set of axioms and an interpreter.

- Axioms constitute a theory of temporal constraint propagation which can be extended even though it already contains the underlying theories of several systems described in [Allen 1981, Le Pape 1985, Rit 1986].
- The interpreter uses the theory consistently with control rules which collectively specify what is expected from the propagation system. Control rules serve as an interface between the problem solver and the interpreter.

A. A Theory of Temporal Constraint Propagation

The basic time objects considered by the propagation system are called "events". Within the theory, each event is identified with an interval of time during which it occurs. This means an event *X* is defined as having only a start point *(start X)* and an end point *(end X).* Symbolic interval-based constraints are used to express various temporal relationships between events (e.g. precedence relation between two events *X* and Y). Numeric point-based constraints are used to represent minimum and maximum distances between time points along a time line (e.g. duration between *(end X)* and *(start Yj).* While propagating symbolic constraints is more efficient and natural, the numeric representation is needed to refine the symbolic relations and take into account lower and upper bounds for durations and waits between events.

Four sources of knowledge make up the whole theory.

- *Logic* axioms concern the manipulation of formulas containing disjunctions, conjonctions and negations. They constitute a variant of propositional logic.
- *Symbolic* axioms deal with the thirteen relations of Allen [Allen 1981].
- *Numeric* axioms deal with the manipulation of inequalities in totally ordered Abelian groups.
- *Translation* axioms enable to convert symbolic constraints **(e.g. *(X before Y) OR (X after Y)*)** into numeric constraints (e.g. ***((end X) < (start Y)) OR ((end Y) < (start X)))***. The converse is also possible, but the numeric information about distances is lost.

An inconsistency is detected when the application of axioms leads to the empty clause.

The theory can be used when a model of time is defined in the form of a totally ordered Abelian group*. Each time point is considered as a variable the range of which is the group. The theory is complete: given a set of constraints, it enables to state whether values can be assigned to time points without violating any constraint.

B. The Interpreter

Within the interpreter, several propagation activities are distinguished:

- *Constraint combination* consists in building a new constraint from a set of existing constraints.
- *Subsumption* enables to hide a constraint the satisfaction of which results from the satisfaction of another constraint.
- *Cancellation* enables to remove a constraint.
- *Restoration* consists in reinstating previously subsumed constraints.
- *Reduction* consists in taking from a constraint what concerns a single time point or interval.
- *Rewriting* enables to gather constraints together into conjunctions, to split conjunctions, to write constraints in normal forms and to translate constraints from a symbolic to a numeric representation (and conversely).

The role of the interpreter is to coordinate these activities in accordance with the instructions of the problem solver. Indeed, the problem solver cannot afford to devote too much of its processing time to a highly combinatorial propagation activity (i.e. to use the completeness of the theory to detect every contradiction as soon as it is introduced). Through control rules, the problem solver restricts the use of the theory.

The interpreter proceeds as follows:

- When a constraint is added or removed, it determines which propagation activities can be performed.
- For each considered activity, control rules specify conditions under which axioms can be used. The interpreter determines whether the concerned constraints satisfy these conditions.
- If the conditions are satisfied, axioms are applied and data dependencies are recorded. Dependency links are necessary for removing all the consequences of a cancelled constraint. Moreover, they enable to provide descriptions of detected inconsistencies.

When the empty clause is derived, a description of the detected inconsistency is sent to the problem solver.

Control rules can address constraint features inherent to the problem domain as well as domain independent features. Within the theory, events are merely time intervals and constraints are formulas. Domain independent features relate to these formulas (e g. disjunctive, numeric, symbolic, binary formulas). Within a problem solver, an event is much more than an interval of time: it corresponds to the performance of an action or to the persistence of a fact. Similarly, constraints partially describe a problem solving situation in terms of goals, restrictions, deviations and consequences of decisions. A lot of data are attached to events and constraints. The conditions specified by control rules can refer to this information.

III CONTROLLING THE FLEXIBLE PROPAGATION SYSTEM

Through control rules, the problem solver indicates how to pertinently use the propagation axioms. Setting these rules requires to take into account the following facts:

- A minimum amount of propagation is necessary for the problem solver to work properly.
- The problem solver does not need to be aware of a contradiction when it has no means to deal with it. Indeed, it may not be provided with the ability to react to contradictions that relate to many unmade decisions or events to happen. There is no need to detect these contradictions before the problem solving state is refined enough for reactive decisions to be made.
- When the problem solver focuses on a particular subproblem, it is worth increasing the amount of propagation within this subproblem.
- The problem solver can spend more or less processing time to propagate constraints in order to assess the consequences of a deviation or a decision.

Part of the knowledge that enables to take these facts into account proceeds from the design of the problem solver. The designer knows how the problem solver works. He is able to a priori appraise the usefulness and the importance of some propagation steps. For example, he is led to define a set of compulsory propagations that are absolutely necessary to ensure that the problem solver will propose admissible solutions.

On the other hand, dynamic knowledge about the problem solving state (e.g. focused objects, importance of satisfying real-time constraints, current problem solving policy and strategy) is required. Indeed, the desirability of some propagation steps varies from one problem to the other and during the course of problem solving. For example, the urgency of reacting to a deviation is estimated in real-time. Similarly, the evaluation of a partial solution against problem domain criteria enables to

A constant _timeO_ and five functions _time<, time=, time,_ _time_+ and _time-_ need to be provided. Moreover, discrete and dense models are distinguished: in a discrete model, there is a smallest positive element _timel_ such that _(time< a b)_ is equivalent to _(time (time* a timel) b)._ A slight alteration in the Abelian group allows to look upon a dense model as a discrete model [Le Pape 1985]. It enables our theory to address both discrete and dense models of time.

determine whether - and to what extent - some constraints or events need particular attention.

In order to dynamically take advantage of the flexibility of the propagation system, the problem solver must be provided with sources of knowledge (e.g. adapting Riles) that perform the following tasks:

- Determining which restrictions must be applied to constraint propagation: the two kinds of knowledge mentioned are considered to determine appropriate restrictions. Knowledge about the efficiency of the propagation system in various kinds of situation is also taken into account. This knowledge can be refined when experience is gained from using the problem solver.
- Accordingly modifying, adding or deleting conditions of control rules.

Obviously, there is a connection between the determination of propagation conditions and other control activities performed by the problem solver (e.g. problem decomposition, adoption of a strategy). The use of control knowledge is extended to the management of the propagation system.

Dynamically controlling the propagation system is helpful in a variety of situations. We present a few examples in the context of job-shop scheduling. Broadly speaking, the scheduling problem consists in assigning resources and start and end times to operations. Operations are the considered events. Our scheduler is the considered problem solver.

- *Considering or neglecting preference constraints:*
  Preferences are constraints the satisfaction of which may be compromised if necessary (e.g. due-dates). The scheduler often relies on the propagation system to detect necessary compromises. However, in case of emergency, the scheduler is only looking for an admissible solution; the propagation system is used as an "admissibility checker": control rules specify that deductive activities (i.e. *combination, reduction, rewriting)* involving preference constraints must not be performed.
- *Focusing on particular events:*
  Propagating disjunctive constraints is time consuming and generally avoided. However, it is worth taking a closer look at operations that require the use of a scarce resource. When scarce resources are identified, the scheduler specifies that disjunctive constraints need to be *combined* only if the associated events require the use of such a resource.
- *Requiring details:*
  Within the scheduler, manufacturing plans are represented as hierarchies of operations (as in [Le Pape and Smith 1987]). Different levels at which scheduling can be performed are defined and explicitely associated with operations. Disjunctive constraints relating to the use of shared resources are more numerous and precise at lower levels. According to the problem solving state, these details about resource planning may be more or less important (e.g. the scheduler may be scheduling in detail or just determining whether due-dates are likely to be relaxed). Consequently, the scheduler specifies levels to which events must belong for propagation to be performed.
- *Adapting the propagation to real-time constraints:*
  In case of emergency, reactive decisions must be made without evaluating all of their consequences. Through control rules, the scheduler specifies that only the consequences that relate to "imminent" events need to be detailed.

IV RESULTING BENEFITS AND REMAINING DIFFICULTIES

With our flexible propagation system, the trade-off between the anticipation of conflicts and the amount of processing time assigned to constraint propagation is not fixed once for all. Through control rules, the amount of propagation is adjusted to meet the needs of a problem solver. Moreover, the problem solver can combine various available pieces of knowledge to dynamically determine how to adjust the behavior of the propagation system.

The problem solver is provided with the ability to moderate its control decisions. Indeed, a wide range of intermediate possibilities from purely heuristic search methods to least-commitment approaches is available. Furthermore, the problem solver can globally adopt a heuristic strategy and ask the propagation system to locally assess all the consequences of a decision. Similarly, it can adopt a least-commitment approach and punctually give up complete consistency checking. This is particularly useful in real-time problem solving. Indeed, the problem solver can adjust the amount of propagation according to the urgency of a reactive decision.

Difficulties in using our flexible propagation system must not be overlooked.

- When a rapid reaction is necessary, the problem solver reduces the amount of propagation. Since the consequences of reactive decisions are not closely considered, new deviations are likely to happen. Deviations may occur more frequently, in which case the propagation activity is reduced again and so forth. Finally, the problem solver might superficially use the propagation system and continually react to unanticipated events.
- The quality of the dynamic adjustment of the amount of propagation rests upon a good understanding of the current problem solving context. This means the problem solver must combine criteria of judgement The difficulty lies in finding relevant criteria as much as in considering them together. For example, with regard to the problem mentioned above (instability resulting from a drastic reduction of the propagation activity), we have no intuition concerning knowledge that enables to evaluate to what extent a set of control rules responds to the need for stability. Furthermore, knowledge would still be required about how to establish a compromise between this need for stability and a need for a rapid reaction.

These difficulties are often encountered (in different forms) in the domain of real-time problem solving. Using the flexible propagation system in the context of job-shop scheduling leads us to investigate domain dependent and independent knowledge that can be used to face these difficulties.

REFERENCES

Allen, J. F. "Maintaining Knowledge about Temporal Intervals." Communications of the ACM 26:11 (1983) 832-843.

Le Pape, C. "Representation et satisfaction de contraintes a l'aide de graphes de potentiel," Laboratoires de Marcoussis, 1985.

Le Pape, C. and S. F. Smith, "Management of Temporal Constraints for Factory Scheduling," Technical Report, Robotics Institute, Carnegie Mellon University, forthcoming.

Rit, J.-F. "Propagating Temporal Constraints for Scheduling." In Proc. AAAI-86. Philadelphia, Pennsylvania, 1986.