Metric Constraint Satisfaction with Intervals

Peter B. Ladkin[1]

TR-89-038

June, 1989

Abstract

We show how algorithms in Dechter, Meiri and Pearl's recent paper on constraint satisfaction techniques for metric information on time points *[DeMePe89]* may be adapted to work directly with metric contraints on intervals. *Inter alia* we show termination of path-consistency algorithms if range intervals in the problem contain only rational number endpoints.

1 International Computer Science Institute, Berkeley, California, on leave from Kestrel Institute, Palo Alto, California.

1 Introduction

In *[DeMePe89]*, Dechter, Meiri and Pearl (henceforth DMP) consider a class of constraint satisfaction problems in which the variables are interpreted as having values from the real numbers, and the constraints contain information on the differences between variables. For them, a constraint on two variables is of the form

$$x_i - x_j \in \bigcup_{1 \leq k \leq m} [a_k, b_k]$$

where for each k, the closed interval $[a_k, b_k]$ with $a_k < b_k$ is the set of real numbers $\{r : \ a_k \leq r \leq b_k\}$, and m depends on i and j. They showed how constraint networks involving convex interval range constraints could be solved in low poynomial time (the case $m = 1$ for each i and j). They indicated that for the general problem, certain approximation (terminology due to *[vBeCoh89]*) or relaxation techniques were applicable, such as path-consistency computations. They noted as two problems that they were unable to show termination of path-consistency computations, and they were unable to characterise all metric constraints on intervals since all the constraints are binary, and some important contraints on intervals are 4-ary on the endpoints.

In this paper, we extend this work by showing how metric information on points and qualitative information on intervals may be combined into the same network. The key issue is passing information on consistency back and forth between the two kinds of information in the network. Complete success is bound to elude us, for we conjecture that the passage of complete consistency information between the metric and the qualitative interval facets of a combined network is NP-hard. However, approximation algorithms used for each facet separately may be used to some effect. We shall assume familiarity with standard notions of boolean constraint satisfaction in general, as contained in e.g. *[Mon74, Mac77, Mac87]*, and with temporal interval constraint satisfaction in particular e.g. *[All83, VilKau86, LadMad88.2]*.

We have shown that qualitative interval networks may be solved equally well no matter whether the endpoints of intervals are from the rational or the real domain (e.g. *[LadMad89.1]*). For computational purposes, it seems that the rational numbers make a more suitable domain of interpretation since they are countable, but have identical first-order order properties to the reals (in qualitative networks, one is only concerned about order properties of endpoints of intervals). When using metric networks, therefore, it makes sense to ask also whether we may restrict ourselves to interpretations over the rational numbers. Since the rational numbers form a field, indeed the smallest field containing the integers and since field operations are the only operations considered by DMP, we may consider their techniques used over rational points. One important result of this is that termination of path-consistency computations now becomes provable, which we show in the next section.

We consider a *metric interval network* to be a network on variables $x_1, ..., x_n$ that have interval values. Associated with each x_i is a pair of variables f_i, g_i that will take values constrained to be the left endpoint, respectively the right endpoint, of x_i. Each pair of variables x_i, x_j has associated with it two sorts of constraints: a qualitative constraint of the sort considered by Allen, and a set of up to four metric constraints between pairs of the variables f_i, f_j, g_i, g_j of the sort considered by DMP. Our approach is to consider how order information may be obtained from solution techniques applied to the metric information, which may then be added in to the qualitative interval constraints. The qualitative interval constraints may then be solved, or approximations generated, and the resulting qualitative information may then be used to constrain the metric information still further. We conjecture that calculation of complete ordering information from solution of even special types of metric network (such as the convex constraint networks solved by DMP) is NP-hard. It is known that solution of general qualitative constraint networks on intervals is NP-hard *[VilKau86]*. So, since a general solution to the problem is infeasible (modula our conjecture), we investigate combinations of techniques and special properties of networks that allow easy solution techniques.

2 Metric Constaint Networks

In this section, we introduce the metric constraint networks of DMP.

A constraint on two variables is of the form

$$x_i - x_j \in \bigcup_{1 \leq k \leq m} [a_k, b_k]$$

where for each k, the closed interval $[a_k, b_k]$ with $a_k < b_k$ is the set of real numbers $\{r : \ a_k \leq r \leq b_k\}$, and m depends on i and j. We refer to the pair of variables $\langle x_i, x_j \rangle$ as the **domain** of the constraint, and the set of real numbers $\bigcup_{1 \leq k \leq m} [a_k, b_k]$ as the **range**. We systematically (try to) use the notation P_{ik} for the constraint with domain $\langle x_i, x_k \rangle$ and similarly use R_{ik} for the range. Thus P_{ik} is the constraint expressed by the formula $(x_i - x_k \in R_{ik})$. We shall denote this by

$$P_{ik} : \ (x_i - x_k \in R_{ik})$$

The constraint says that the difference between the values of the two variables x_i and x_j is to lie in the union of the convex closed intervals $[a_k, b_k]$. We shall indicate the domain of the constraint by the usual syntactic notation for formulas, i.e. $P_{ik}(x_i, x_k)$, where we need to.

A **form** of a range is a set $\{\langle a_k, b_k \rangle : 1 \leq k \leq m\}$ where $R = \bigcup_{1 \leq k \leq m} [a_k, b_k]$. A range thus has many forms, because there is no criterion that the intervals represented in a form be disjoint. We define the **normal form** of a range to be a form $\{\langle a_k, b_k\rangle : 1 \leq k \leq m\}$ where for $1 \leq k < m$, $a_k < b_k < a_{k+1}$. We define $NF(R)$ to be the normal form of R. It is immediate that ranges are in one-to-one correspondence with their normal forms. We shall often express operations on ranges by giving operations on their normal forms. A range R' is a **subrange** of the range R if $R' \subseteq R$. Expressed in terms of normal forms, it is easy to check that this is equivalent to the following condition. If $NF(R) = \{\langle a_k, b_k\rangle : 1 \leq k \leq m\}$ and $NF(R') = \{\langle c_j, d_j\rangle : 1 \leq j \leq p\}$, then for each $\langle c_j, d_j\rangle \in NF(R')$, there exists $\langle a_k, b_k\rangle \in NF(R)$ such that $a_k \leq c_j \leq d_j \leq b_k$. Thus we may consider set-type operations on ranges to be equivalent to algebraic operations on their normal forms. This is the approach taken by, for example, the relation algebra and cylindric set algebra of Tarski (e.g. *[Tar41, HeMoTa??]*), and by the relational algebraic approach to relational data bases, where it has proved fruitful. We have used relation algebra to characterise operations on qualitative interval networks in *[LadMad89.1, LadMad88.2]*.

Associated with the constraint $P_{ik}(x_i, x_k)$ is the binary relation Q_{ik}:

$$Q_{ik} = \{\langle a_i, a_k\rangle : (a_i - a_k \in R_{ik})\}$$

which we call the **satisfaction set** of the constraint P_{ik}. Since one has to know merely the variables involved, and the acceptable pairs of values of those variables to known a constraint, the constraint P_{ik} is specified precisely by its domain along with the satisfaction set, so a constraint may be considered as a triple $\langle i, k, Q_{ik}\rangle$. We extend the notion of range to include the **range** of a satisfaction set, defined as R, where $Q = \{\langle a_i, a_k\rangle : (a_i - a_k \in R)\}$. We will denote the range of Q as $R(Q)$ in the case where the set Q is not associated with any particular constraint. For an arbitrary set of pairs of real numbers Q, $R(Q)$ is just a set of real numbers.

An n-ary constraint network is a collection of these pairwise constraints on a set $x_1, \ldots\ldots, x_n$ of n variables, along with unary constraints (as below). Thus an n-ary constraint network may express up to $n(n-1)$ binary constraints. Since $x_i - x_i = 0$, we have to define unary constraints differently. A **unary constraint** P_i on a variable x_i is a constraint of the form $x_i \in R_i$, where R_i is a range as defined above. The network is **satisfied** by an assignment of real numbers $a_1, \ldots, a_n$ to the variables $x_1, \ldots, x_n$ such that $a_i - a_j \in R_{ij}$ for each binary constraint P_{ij} (i.e. if we consider P_{ij} to be a formula with free variables x_i, x_j, then a_i, a_j satisfy this formula), and $a_i \in R_i$ for each unary constraint P_i. A network may be represented in the usual way (e.g. *[Mac87]* as a labelled digraph in which the nodes represent the variables, and the edges are labelled with the constraint on the variables corresponding to the nodes at the head and tail. DMP give examples of such constraint networks and their solution.

We shall define next the usual notions associated with the solution of constraint satisfaction problems, for the case of metric networks. Thus techniques from *[LadMad.88.2]* may be directly adapted to networks of these constraints. We shall also show that the computation of path-consistency for rational number interpretations always terminates.

3 Operations on Metric Constraints and Networks

In order to perform operations on networks of constraints, we shall need to define certain operations on constraints. We shall then use these operations to define prunings, or reductions, of the constraints in a network, as an aid in satisfying (i.e. finding solutions to) the network. In particular, as for DMP, we shall concern ourselves with the notion of path-consistency. In order to apply algebraic techniques and algorithms from *[LadMad88.2]*, we need to define the **sum**, or union; the **product**, or intersection; the **relative product**, or composition, and the **converse** of constraints.

The definitions of union, intersection and composition are due to DMP. The **sum** and **product** of two binary constraints, $P_{ij} + P'_{ij}$ resp $P_{ij} \cdot P'_{ij}$, are given by the formulas

$$P_{ij} + P'_{ij} \ : \ x_i - x_j \in R_{ij} \cup R'_{ij}$$

respectively

$$P_{ij} \cdot P'_{ij} \ : \ x_i - x_j \in R_{ij} \cap R'_{ij}$$

i.e. the sum and product are given by the union and intersection of the ranges. It is easily checked that the satisfaction sets of the sum and product of two constraints are respectively the union and intersection of the satisfaction sets of the constraints.

We define the **composition** of the ranges $R \circ S$ as follows;

$$\alpha \in (R \circ S) \Leftrightarrow (\exists\beta)(\exists\gamma)(\beta \in R \ \wedge \ \gamma \in S \ \wedge \ \alpha = \beta + \gamma)$$

The composition of ranges is defined for every pair of ranges. In contrast, we will define the **composition** of two constraints only between constraints that have a variable in common, say x_j. Let (P_{ij}, P_{jk}) be two constraints with the common variable x_j. The composition constraint $(P_{ij} \circ P_{jk})$ is defined as follows:

$$(P_{ij} \circ P_{jk}) : \ x_i - x_k \in (R_{ij} \circ R_{jk})$$

The importance of the notion of composition is that $x_i - x_j \in R_{ij}$ and $x_j - x_k \in R_{jk}$ imply that $x_i - x_k \in (R_{ij} \circ R_{jk})$, as may be easily checked from the defining formula. This means that the information contained in the composition constraint on x_i, x_k is only the information derived from knowing the constraints on the pairs x_i, x_j and

x_j, x_k. $(R_{ij} \circ R_{jk})$ may be understood as the narrowest range to which $x_i - x_k$ must belong, given only the constraints P_{ij} and P_{jk}. The definition of composition is derived from the notion of the composition of two binary relations. Specifically, the standard definition of the composition of two binary relations *[LadMad88.2]* is

$$S_1 \circ S_2 \;=\; \{\langle a, c\rangle : (\exists b)(\langle a, b\rangle \in S_1 \;\wedge\; \langle b, c\rangle \in S_2\}$$

and it is easily checked that the satisfaction set of the composition of two constraints is the composition of the satisfaction sets of each constraint.

Converse: We shall need the notion, which DMP do not use, of the **converse** of a constraint. Given a constraint P_{ij}, the converse is denoted by $(P_{ij})^{\smile}$. The domain of $(P_{ij})^{\smile}$ is $\langle x_j, x_i\rangle$, and represents the 'same' constraint as (P_{ij}), in the sense that the logical force of the converse constraint is exactly that of P_{ij} itself, but the variables appear in the reverse order. To motivate the definition, consider the case where $R_{ij} = [a, b]$, i.e. the range is a single convex interval. Since $a \leq x_i - x_j \leq b$, then reversing the signs gives $-b \leq x_j - x_i \leq -a$. So first we define the **converse** of an interval $[a, b]^{\smile}$ to be the interval $[-b, -a]$, and the converse of a range $R_{ij}{}^{\smile}$ as the union of the converses of the component intervals, i.e. if

$$R_{ij} = \bigcup_{1 \leq k \leq m} [a_k, b_k]$$

then

$$R_{ij}{}^{\smile} = \bigcup_{1 \leq k \leq m} [a_k, b_k]^{\smile} = \bigcup_{1 \leq k \leq m} [-b_k, -a_k]$$

We may now define the **converse** of a constraint

$$P_{ij} :\; x_i - x_j \in R_{ij}$$

to be the constraint

$$P_{ij}{}^{\smile} :\; x_j - x_i \in R_{ij}{}^{\smile}$$

It should be clear that the pairs of values of x_i and x_j satisfying both P_{ij} and $P_{ij}{}^{\smile}$ are the same. This may be stated also in terms of satisfaction sets. As in relation algebra, we define the **converse** $S^{\smile}$ of a set of pairs S by:

$$S^{\smile} \;=\; \{\langle b, a\rangle : \langle a, b\rangle \in S\}$$

It follows that the satisfaction set of $P_{ij}{}^{\smile}$ is $S_{ij}^{\smile}$. (This can be thought of in the following way: if we reverse the order we consider the variables in, i.e. the domain, then we have also to reverse the order we consider satisfying instances in, i.e. we need the converse of the satisfaction set).

We define a network A' to be a **reduction** of a network A if (a) A and A' have exactly the same variable sets, and for each constraint P_{ij} of A there is a corresponding constraint P'_{ij} of A' such that $R'_{ij} \subseteq R_{ij}$. We say that A' is an **stable reduction** of A (also called a *relaxation* in *[DecPea88]*) if and only if A' is a reduction of A, and A and A' have exactly the same n-tuples of global solutions, i.e.

$$\{\langle a_1, ..., a_n\rangle : (\forall i \neq j \leq n)(a_i - a_j \in R_{ij}) \ \wedge \ (\forall i \leq n)(a_i \in R_i)\}$$
$$=$$
$$\{\langle a_1, ..., a_n\rangle : (\forall i \neq j \leq n)(a_i - a_j \in R'_{ij}) \ \wedge \ (\forall i \leq n)(a_i \in R'_i)\}\}$$

and also that Much of the work in constraint satisfaction is concerned with computing reductions and stable reductions of networks. Usually, concepts such as arc- and path-consistency are used to compute stable reductions, and then search techniques are used to compute (non-stable) reductions of these (c.f. *[LadMad88.2]*).

3.1 Regular Metric Networks

The concept of a regular network was defined in *[LadMad88.3]*, and this applies to metric networks, given the definition of composition above. A network is **regular** if there is only one constraint between any given pair of variables x_i and x_j. A regular network may be represented as a digraph with a single directed edge between a given pair of nodes. For any network there is an easily computable regular network which is a stable reduction. Suppose a nonregular network has both constraints $P_{ij}(x_i, x_j)$ and $P_{ji}(x_j, x_i)$ for some pairs of variables. The following algorithm computes the **most general regular reduction** (which is also stable). (A most general regular reduction is a regular reduction A' such that any regular reduction A'' of A is also a reduction of A').

Regular Reduction Algorithm For every pair of variables x_i, x_j with $i < j$, replace the pair of constraints $P_{ij}(x_i, x_j)$ and $P_{ji}(x_j, x_i)$ with the single constraint $P'(x_i, x_j)$ where P' is the constraint $(P_{ij} \cap P_{ji}^{\smile})$.

(This algorithm is from *[LadMad88.3]*). The justification for this algorithm is as follows. As we noted above $P_{ji}(x_j, x_i) \equiv (P_{ji})^{\smile}(x_i, x_j)$. So both P_{ij} and $P_{ji}{}^{\smile}$ give a range for $x_i - x_j$, and therefore any value for $x_i - x_j$ is constrained to lie in both the ranges R_{ij} and $R_{ji}{}^{\smile}$, and therefore in the intersection. The new network is regular, and has the same set of solutions as the original network, as should be easy to see. Henceforth we shall assume that the networks are all regular, and use the conventions that P_{ij} appears as a constraint if $i < j$, and that the constraint with domain $\langle x_j, x_i\rangle$ with $j > i$ is $(P_{ij})^{\smile}$. We shall say, for regular networks, that the domain of a constraint is the set $\{x_i, x_j\}$ without fear of ambiguity.

3.2 Arc-consistency in Metric Networks

We shall need to define the *binarification* of a unary relation. Suppose R is a unary relation on a set U. We define the **binarification** R^b to be that relation on $(U \times U)$ defined as

$$R^b(i,i) \Leftrightarrow R(i)$$

For any relation R, R^b is a subrelation of the identity relation $Id(U)$, i.e. $R^b \subseteq Id(U)$. For a constraint network A, define $R_{ii} = (R_i)^b$. (This was the approach taken to coercing unary constraints to binary constraints in *[LadMad88.2]*). Using the definitions above, we may now define a metric network to be **arc-consistent** if and only if

$$\forall i \neq j \leq n \ : \ \pi_1(S_{ij}) = R_{ii} \ \wedge \ \pi_2(S_{ij}) = R_{jj}$$

where π_1, π_2 are the projection functions on respectively the first and second co-ordinates of a set of pairs. Thus a network is arc-consistent if and only if every value specified by a unary constraint on a variable is also consistent with all binary constraints on that variable, i.e is part of a value-pair in the satisfaction set of every binary constraint on that variable. Notice if the network is regular, then the second clause is redundant. Another formulation of arc-consistency, which may be found in *[LadMad88.2]*, is the following:

$$\forall i \neq j \leq n \ : \ R_{ii} \circ S_{ij} = R_{jj}$$

Recent work on another algebraic formulation of arc-consistency constraints, with some algorithms, may be found in *[GüsHer88]*, where it is called *local consistency*.

3.3 Path-consistency

Given these definitions, we may define **path-consistency** of a network as follows. The definitions and algorithm are adapted from *[LadMad88.2]*. A network is **path-consistent** if and only if

$$R_{ij} \subset (R_{ik} \circ R_{kj})$$

for every $i, j, k \leq n$. This could also be stated in terms of the satisfaction sets: $S_{ij} \subset (S_{ik} \ \circ \ S_{kj})$ for every $i, j, k \leq n$. The following algorithm scheme computes the **most-general path-consistent reduction** $MGR_3(A)$ of a network A. (A **triangle** (i, j, k) is the triple of variables $\langle x_i, x_j, x_k \rangle$.)

Path Consistency Algorithm: Given a network A, iterate until no more changes: For every triangle (i, k, j) in A: do $P_{ij} \leftarrow P_{ij} \cdot (P_{ik} \ \circ \ P_{kj})$.

DMP showed that execution of this algorithm may be accomplished by the Floyd-Warshall all-pairs-shortest-paths algorithm for the case of metric constraint networks. We showed in *[op.cit.]* that path-consistency may be computed in time

$O(n^2 logn)$, and that there are examples that will cause standard types of algorithms (*reduction*-type algorithms *[op.cit]*), whether parallel or serial, to take time $\Omega(n^2)$, thus providing effectively an asymptotic lower-bound for such computations.

3.4 Termination of Path-Consistency

Here we show that path-consistency computations always terminate for the case where the ranges of the constraints have endpoints from the rational numbers. This will suffice for our concerns, since we work with the countable model of Allen's interval algebra, where the endpoints are rationals *[LadMad89.1]*. In particular, we are ultimately interested in the *TUS* model of the interval algebra, which is isomorphic to the pairs-of-rationals model *[Lad88.1]*.

First, we show that path-consistency computations for integer ranges terminate.

Lemma 1 *Suppose A is an n-ary constraint network in which all the ranges R_{ij}, $i,j \leq n$ have integer endpoints; i.e. if $NF(R_{ij}) = \{\langle a_k, b_k\rangle : 1 \leq k \leq m\}$ then for $1 \leq k \leq m$, a_k, b_k are integers. Then the path-consistency computation scheme above terminates.*

Proof: We use the standard notation $\mathbf{Z}$ for the integers. Let $max(R)$, $min(R)$ be respectively the largest, smallest integers appearing in a convex interval in R, and let max, min be the largest, respectively the smallest, of the $max(R), min(R)$. Let the **scope of** R, $SC(R) = \{a \in \mathbf{Z} : min(R) \leq a \leq max(R)\}$. Let the **scope of** A, $SC(A) = \{a \in \mathbf{Z} : min \leq a \leq max\}$. Note $SC(R) \subseteq SC(A)$ for R a range in a constraint in A. Let $EP(R)$ be the set of endpoints of intervals in $NF(R)$, so $EP(R) = \{a_1, b_1, a_2, \ldots\ldots, a_m, b_m\}$, where $R = \bigcup_{1\leq k\leq m}[a_k, b_k]$ and $a_1 < b_1 < a_2 < \ldots\ldots < a_m < b_m$. Let $Ranges(A) = \{\ R : R \text{ is a range, and } EP(R) \subseteq SC(A)\}$. Then $Ranges(A)$ is a finite set, since $SC(A)$ is finite, hence has finitely many subsets, and the $EP(R)$ are in one-to-one correspondence with ranges R. Note also that $Ranges(A)$ is closed under subranges.

Note the following truths: $EP(R) \subseteq SC(R)$, $EP(R \cup S) \subseteq (EP(R) \cup EP(S)) \subseteq (SC(R) \cup SC(S))$, $EP(R \cap S) \subseteq (EP(R) \cup EP(S))$, and $EP(R \cap S) \subseteq (SC(R) \cap SC(S))$. From these it easily follows that $Ranges(A)$ is closed under the operations of union and intersection.

If R is a range in A, and S is any integer range, then $EP(R \cap S) \subseteq (SC(R) \cap SC(S)) \subseteq SC(A)$. It may be easily checked that the definition of composition preserves integer endpoints. Consider now the operation involved in the path-consistency operation, rephrased in terms of ranges, $R_{ij} \leftarrow R_{ij} \cdot (R_{ik} \circ R_{kj})$. The result is a subrange of R_{ij} (possibly including many more convex subintervals), therefore still in the set $Ranges(A)$. $Ranges(A)$ is a finite set, so the partial order

under the relation of subrange is well-founded. Since the path-consistency operation at each instantiation produces a subrange, the operation above must terminate for every i, j, k. Hence the path-consistency computation terminates.
End of Proof.

Lemma 2 *Suppose A is an n-ary constraint network in which all the ranges R_{ij}, $i, j \leq n$ have rational number endpoints; i.e. if $NF(R_{ij}) = \{\langle a_k, b_k \rangle : 1 \leq k \leq m\}$ then for $1 \leq k \leq m$, a_k, b_k are rationals. Then the path-consistency computation scheme terminates.*

Proof: Let $EP(A) = \bigcup_A EP(R)$, i.e. the collection of endpoints of ranges in A. Given a range $R = \bigcup_{1 \leq k \leq m}[a_k, b_k]$, let $m * R = \bigcup_{1 \leq k \leq m}[(m \times a_k), (m \times b_k)]$. The following distributive laws are easy to check: $m * (R \cup S) = (m * R) \cup (m * S)$, $m * (R \cap S) = (m * R) \cap (m * S)$, $m * (R \circ S) = (m * R) \circ (m * S)$.

Let $M = gcd(EP(A))$, and let $(M * A)$ be the network just like A except every range R in each constraint has been replaced by $(M * R)$. $(M * A)$ is a network with integer ranges, and hence path-consistency computations terminate, by the previous lemma. By the distributive laws above, the path-consistency computation on $(M * A)$ is isomorphic to the computation on A, since each operation commutes with the factor M. Hence the computation on A also terminates.
End of Proof.

We are mainly concerned with countable interval models of Allen's system (which are all isomorphic *[LadMad89.1]*), and in particular with the TUS, so these results suffice for us to show termination of path-consistency. The argument of the lemma cannot be extended to even the case of normal forms with algebraic number endpoints, since the set of endpoints obtainable by performing addition and subtraction operations on the endpoints is now no longer finite within the interval $[min, max]$. We need not be concerned with this unless we have a real need for real numbers (one might even regard this lemma as providing a reason why real numbers should be avoided here).

4 Interval Constraint Networks with Metric Information

DMP have developed constraint satisfaction techniques for metric information using the computations on the constraint structures described above. Our concern here is to adapt these techniques for use with interval constraint networks. DMP and others have pointed out that many common interval relations cannot be phrased in terms of binary constraints upon the endpoints of the intervals *[Mei89]*. For example, the relation of disjointness between two intervals $[a, b]$, and $[c, d]$ is expressed by

$b \leq c \ \vee \ d \leq a$ and there seems to be no simple way of phrasing this constraint which involves only two of the endpoints. DMP may consider it a weakness of their approach that it does not appear to handle those interval relations that are not expressible with only two endpoint variables. We overcome some of the problems by devising constraint networks that have constraints of two types. One type of constraint is the qualitative interval constraint of Allen, and overlaid on top of these is another set of constraints containing metric information of the sort DMP consider, but on endpoints of the intervals.

4.1 Metric Interval Networks

We define a **metric interval network**, or MIN, as a constraint network where the constraints P_{ij} are pairs of constraints $\langle S_{ij}, T_{ij}\rangle$ where S_{ij} is a metric constraint involving four possible temporal ranges as explained below, and T_{ij} is an Allen-type interval constraint of the sort considered in *[All83, LadMad88.2, LadMad89.1]*. The constraints S_{ij} are combinations of metric constraints (as in DMP) between the endpoints of intervals x_i and x_j. We use the variables f_j, g_j for the left and right endpoints of the interval denoted by x_j, as in *[Lad88.1]*. (Other logical notation in this section is also as in *[Lad88.1]*.) S_{ij} is then a conjunction of some or all of the constraints

$$(f_i - f_j \ \in R^1{}_{ij}) \ \wedge \ (f_i - g_j \ \in R^2{}_{ij}) \ \wedge \ \ (g_i - f_j \ \in R^3{}_{ij}) \ \wedge \ (g_i - g_j \ \in R^4{}_{ij})$$

i.e. any subset of these conjuncts may appear as the constraint S_{ij}.

Let A be a MIN. Then the S_{ij} form a DMP-type metric network with $2n$ variables $f_1, g_1, f_2, g_2, ..., f_n, g_n$ with the range constraints as specified by the $R^k{}_{ij}$. We call this network the **associated metric network** of A, or MN(A). We refer to an MN(A) where all the $R^k{}_{ij}$ are convex intervals as a **convex** MN(A), and a MIN with a convex MN(A) as a **convex MIN**. For the case of convex MINs, DMP show how the MN(A) may be constructed in $O(n^3)$ time, by using the Floyd-Warshall all-pairs-shortest-paths algorithm. Similarly, there is an associated qualitative interval network with n variables obtained by considering only the constraints T_{ij}. We call this the *associated Allen network*, or AN(A). For cases where the AN(A) is *pointisable [LadMad88.2]*, it may be solved in serial $O(n^3)$, or parallel $O(n^2 \ log\, n)$ time, by applying a path-consistency algorithm, and then applying an $O(n^2)$ satisfaction algorithm to the resulting path-consistent network (path-consistency is equivalent to satisfiability for pointisable AN(A)s, although not for general AN(A)s) *[LadMad88.2]*.

We say that a MIN is **satisfied** by a sequence $\langle l_1, r_1, l_2, r_2, ..., l_n, r_n\rangle$ if the assignment $[f_1 \leftarrow l_1, g_1 \leftarrow r_1, f_2 \leftarrow l_2, g_2 \leftarrow r_2, ...f_n \leftarrow l_1, g_n \leftarrow r_1]$ satisfies the MN(A), and the assignment $[x_1 \ \leftarrow \ \langle l_1, r_1\rangle, x_2 \ \leftarrow \ \langle l_2, r_2\rangle, ...x_3 \ \leftarrow \ \langle l_3, r_3\rangle]$ satisfies the AN(A).

4.2 Solution Techniques For Convex MINs

In this section we consider solving convex MINs. DMP show how to solve convex MN(A)s, by calculating the minimal network associated with a given convex MN(A). There are also techniques for approaching the solution of AN(A)s, although only in some cases (namely pointisable AN(A)s) is it known that a solution may be found in low polynomial time *[LadMad88.2]*. The major issue for MINs is how to integrate the solution techniques of both MN(A)s and AN(A)s so that inconsistencies between the qualitative and quantitative sides of the MIN may be discovered. Here, we formulate the problems and the options, and indicate how one may obtain polynomial time approximations to solutions. We conjecture that calculating complete information from both sides is NP-hard in general.

For any convex MN(A) T, let $MNR(T)$ be the minimal network representation of T (as defined in DMP). The minimal network representation in this case has the property that any partial solution (i.e. an assigment to some of the variables that satisfies the constraints on those varaibles alone) can be extended to a complete solution of the network. In particular, each of the domains (unary constraints) is feasible, that is contains precisely values of the variable that may participate in a solution. However, it is not the case that any element of the cartesian product of the feasible domains is a solution of the constraint graph. But it is the case that any value of a domain that is consistent with partial assignments to the previous variables may participate in a global solution which extends that partial assignment, as shown by DMP, and see below.

To illustrate our approach to transferring consistency information between the MN(A) and the AN(A), consider the constraint

$$x_2 - x_1 \in [5, 7]$$

This constraint is equivalent to the conjunction of the two constraints

$$x_2 - x_1 \geq 5$$

$$x_2 - x_1 \leq 7$$

both of which imply the qualitative constraint

$$x_2 \geq x_1$$

On the other hand, a constraint of the form

$$x_2 - x_1 \in [-1, 1]$$

may be factored into

$$x_2 - x_1 \geq -1$$

$$x_2 - x_1 \leq 1$$

which is consistent with both of the qualitative constraints

$$x_2 \geq x_1$$

$$x_2 \leq x_1$$

and so provides no qualitative information on the ordering of x_1 and x_2. The proof of minimality by DMP shows that we may pick any value for x_1 satisfying the domain constraint on x_1, and then any value for x_2 satisfying the constraint $x_2 - x_1 \in [-1, 1]$, and may then extend this assignment to a solution. Hence there are solutions of the MN(A) in which $x_2 \geq x_1$ and solutions in which $x_2 \leq x_1$. However, choice of one of these qualitative orderings for x_1 and x_2 might affect choices of values for later variables and therefore could constrain the ordering of later variables beyond what the explicit ordering in the minimal network demands. So not all these sequential choices may be independent of previous ones. This leads to the following definition.

We define the **qualitative constraint form** $QC(A)$ of an MN(A) A to be the disjunction of all orderings $x_{i_1} R x_{i_2} \ \wedge \ .... \ \wedge \ x_{i_{n-1}} R x_{i_n}$, where R is either $<$, $=$ or $\leq$, which are the ordering of a solution to A. (As a shorthand, we may write such an ordering as $x_{i_1} R x_{i_2} R x_{i_3} R .... R x_{i_{n-1}} R x_{i_n}$). The object is to compute a formula equivalent to $QC(A)$ for A the MN(A) of a MIN T, and then to combine this information with the AN(A), using the tramslation methods between interval formulas and point formulas of *[Lad88.1]*. However, as we have noted, $QC(A)$ may be hard to compute.

The technique suggested by the example given above doesn't compute $QC(A)$ as we noted. It computes an approximation to $QC(A)$ which we make precise with the following definitions. We define the **associated constraint** y_{ij} of a metric constraint $x_i - x_j \in [a, b]$ to be $x_i > x_j$ if a is positive; $x_i \geq x_j$ if a is 0; $x_i < x_j$ if b is negative; $x_i \leq x_j$ if b is 0; and NIL if a is negative and b is positive. We define the **weak qualitative constraint form** $WQC(A)$ to be the conjunction

$$y_{21} \ \wedge \ y_{31} \ \wedge \ ......y_{n1} \ \wedge \ y_{32} \ \wedge .... \wedge \ y_{n,n-1}$$

(where we omit the constraint if is is equal to NIL). $WQC(A)$ is easy to compute. It follows easily from the definition that it is $O(e)$, where e is the number of edges in the network.

Clearly, considered as formulas, $QC(A) \Rightarrow WQC(A)$ so $WQC(A)$ may be used as a *filter* for approximating $QC(A)$. A filter is a formula which implies, but is not necessarily implied by, a formula of interest *[Smi88]*. Filters are logically weaker than the formula of interest. Filters are generally easy to compute, whereas the formula of interest ($QC(A)$ in this case) may be relatively hard to compute. They are used in order to prune the problem space. They are effective in so far as they are logically 'not much weaker', and in so far as it is easy to compute them.

$WQC(A)$ is easy to calculate, requiring linear time in the number of constraints, i.e. $O(e)$ where e is the number of edges. There is a translation between endpoint constraints of the form occurring in $QC(A)$ and $WQC(A)$ and interval constraints of the form appearing in $AN(A)(A)$. The translation is given in *[Lad87.5, Lad88.1]*, and we reproduce it below.

4.3 Information Passing From Points to Intervals

However we compute approximations to $QC(A)$, we may translate the computed endpoint constraints into interval constraints, and conjoin them with the constraints on the respective edges in $AN(A)$, using the translation above. We call this new qualitative network an **augmented** $AN(A)$, $AGAN(A)$. This qualitative network may be solved, or approximated, using known techniques such as those in *[All83, VilKau86, Lad 87.5, Lad 88.1, LadMad88.2, vBeCoh89]*. A solution, or approximate solution, of the $AGAN(A)$ involves an *atomic network [LadMad88.2]*, or a disjunction of atomic networks (except for techniques involving the minimal network *[Mon74, vBeCoh89]*). It will be seen from the way that these constraints are used in augmenting the metric constraints that disjunctive constraints are not much help, since they engender combinatorial explosion in the same way that a crude search for atomic reductions of a constraint network does. One approach to this is to restrict attention to certain classes of networks that admit of easier solution methods. We consider this further in a later section.

4.4 Incorporating the Qualitative Solution in the Metric Constraints

Given a solution, approximate solution, or class of solutions to the $AGAN$, we may now translate back the interval constraints on the variables to endpoint constraints using the translation schemes in *[Lad87.5, Lad88.1]*. We give the translation, for completeness, below. These solutions form a filter for solving the $MN(A)$, in the following manner. Atomic networks are pointisable, and so may be translated into conjunctive constraints on endpoints, using the scheme below. Constraints between each pair of endpoints may be calculated from these constraints by using transitivity. The derived qualitative constraint between interval x_i and interval x_j translates into possibly four constraints on the endpoints, which are conjunctive and therefore partially specify a linear ordering of the endpoints. These constraints may then be used to prune the metric constraints on x_i and x_j, in, say, the following way. Recall that the metric constraints on x_i and x_j are a collection of up to four metric constraints involving the four variables f_i, g_i, f_j, g_j. Suppose one of the constraints is, say,

$$f_i - g_j \in [-5, 7]$$

and the derived qualitative constraint on f_i and g_j is $f_i > g_j$. Then the metric constraint may be further pruned to yield

$$f_i - g_j \in (0, 7]$$

which is the part of the metric constraint consistent with the derived qualitative information. If the derived constraint were $f_i < g_j$, then the derived metric constraint would be

$$f_i - g_j \in [-5, 0)$$

Similarly, for a derived qualitative constraint $f_i \geq g_j$, the derived metric constraint would be

$$f_i - g_j \in [0, 7]$$

and *mutatis mutandis* for $f_i \leq g_j$. Although the half-open interval is not exactly of the form considered by DMP, constraint satisfaction involving half-open intervals may be implemented by considering the constraint to be the closed interval $f_i - g_j \in [0, 7]$ and annotating the constraint with the information that $f_i \neq g_j$. The annotation can be incorporated into any satisfaction routine for the metric information, e.g. as in DMP. The routine that DMP use to show that the minimal-path-distance network is indeed the minimal network representation of the metric constraint network builds a solution in the following way. Suppose without loss of generality that values $a_1, \ldots\ldots, a_{i-1}$ for variables $y_1, \ldots\ldots, y_{i-1}$ have been picked, that are consistent with the mutual constraints on $y_1, \ldots\ldots, y_{i-1}$. Any value of y_i may then be used to extend the partial solution, provided it is within the range $R = [max(a_j - d_{ij} : j < i), min(a_j + d_{ji} : j < i)]$. When such a value is picked, one should additionally check all the annotations $y_j \neq y_i$ for $j < i$. Suppose that $y_{i_1}, \ldots, y_{i_k}$ have such annotations. Then any value in the range $R \sim \{a_{i_1}, \ldots, a_{i_k}\}$ may be used for y_i (where $\sim$ is the set-difference operator), and consistently extended to a solution which incorporates the inequality annotations also.

The qualitative information obtained from solving or partially solving the $AGAN$ may be incorporated back into the $MN(A)$ in time $O(e)$, as is easily seen from the exposition above, and the translation scheme below.

5 The Translations

5.1 Definitions and Notation

For the purposes of this section we shall consider qualitative constraints on intervals in a network as formulas in a first-order logical language for the purposes of giving the translations. The language contains a binary relation symbol for each of the fundamental thirteen binary relations on intervals. A complete axiomatisation of a first-order theory which has the 'same' models as the algebraic formulation (in a sense explained there) may be found in *[LadMad89.1]*. So considering constraints as formulas is just another (sometimes convenient) way of looking at the setup, and has provably all the 'same' properties. The disjunctive constraints typically found as edge constraints in a network are obtained by taking disjunctions of some of the thirteen relations. So an edge between, say, x_i and x_j with $i < j$ has a constraint that can be represented in this manner as $P(x_i, x_j)$, and the formula $P(x_i, x_j)$ is equivalent to $G_{k_1}(x_i, x_j) \lor \ldots\ldots \lor G_{k_m}(x_i, x_j)$, where $G_{k_1}, \ldots\ldots G_{k_m}$ are the predicate symbols corresponding to those fundamental relations that are disjoined to form the constraint P_{ij}. We shall employ this method of talking about the constraints throughout our presentation of the translations and information-passing between the metric and qualitative parts of the MIN.

As well as the formulas themselves, we shall need to interpret them over a domain. We use the standard notations of model theory to describe the domains we are interested in (the rational numbers and the intervals over the rationals for the two types of constraints we have). We use notation consistent with that in *[Lad87.5]* and *[Lad88.1]*. We use RAT to denote the structure of the rational numbers with the natural ordering, $\langle \mathbf{Q}, < \rangle$. We use INT to denote the structure of the intervals over the rational numbers with the twelve fundamental relations (equality is the thirteenth) $\langle INT(Q), G_1, \ldots\ldots, G_{12} \rangle$, The domain $INT(Q)$ consists of all pairs of rational numbers $\langle a, b \rangle$ with $a < b$ (the intervals over the rationals). As we use the notation $\langle l, r \rangle$ to denote the interval whose left endpoint is l and whose right endpoint r, we shall use the notation i_L and i_R to denote the left, respectively the right, endpoint of the interval i. Thus it is a truism that $i = \langle i_L, i_R \rangle$.

We shall abuse notation occasionally in referring to the rational numbers $\mathbf{Q}$ when we mean RAT, and the rational intervals $INT(Q)$ when we mean INT. For *assignment*, in defining satisfaction of formulas by sequences of model elements, we use the notation e.g. $[z_1 \leftarrow l_1, z_2 \leftarrow l_2, \ldots\ldots, z_n \leftarrow l_n]$ to denote the simultaneous assigment of the model element l_i to the variable z_i, for $1 \leq i \leq n$, and we say e.g. $\mathcal{M} \models \phi[z_1 \leftarrow l_1, z_2 \leftarrow l_2, \ldots\ldots, z_n \leftarrow l_1]$ to mean that $\mathcal{M}$ satisfies the formula ϕ with free variables amongst $z_1, .., z_n$ under the assignment $[z_1 \leftarrow l_1, z_2 \leftarrow l_2, \ldots\ldots, z_n \leftarrow l_1]$.

The thirteen fundamental interval relations are denoted P, M, O, D, S, F, $P^\smile, M^\smile, O^\smile, D^\smile, S^\smile, F^\smile$ and $Id(L)$ for *precedes, meets, overlaps, during, starts, finishes*, their converses, and the identity relation on $INT(Q)$, respectively (the notation $Id(U)$ denotes the identity relation on a set U, i.e. $\{\langle x, x \rangle : x \in U\}$, and L is the set of intervals on $\mathbf{Q}$, i.e. $\{\langle x, y \rangle : x, y \in \mathbf{Q} \land x < y\}$. Notice conincidentally that L is also the binary relation of $<$ on $\mathbf{Q}$, an element of the point relation algebra on $\mathbf{Q}$) *[All83, LadMad89.1]*.

When defining the translations and noting their properties, we shall use variables f_i, g_i for (intended) right and left endpoints of intervals, and variables e_i for intervals (i.e. in interval formulae). It is intended that the variables f_i, g_i shall correspond to the right, respectively the left, endpoints of the interval variable e_i. So if in particular we state a property of a formula $\phi(f_i, g_i)$ under an assignment $[f_i \leftarrow l, g_i \leftarrow r]$, we shall expect to be concluding things about a 'corresponding' interval statement $\psi(e_i)$ under the assignment $[e_i \leftarrow \langle l, r\rangle]$, and vice versa. This is a piece of bookkeeping that we hope aids the statement and the understanding of the translations.

5.2 From Points to Intervals

We provide a syntactic translation from formulae in the language of points to formulae in the language of intervals, that preserves satisfaction in the following sense. Consider a formula ϕ in the language of the rational numbers with order, $\langle \mathrm{Q}, < \rangle$, in the variables $f_1, g_1, .., f_n, g_n$, and suppose the assignment $[f_1 \leftarrow l_1, g_1 \leftarrow r_1, f_2 \leftarrow l_2, g_2 \leftarrow r_2, ...f_n \leftarrow l_n, g_n \leftarrow r_n]$ satisfies the formula ϕ over the rational numbers (i.e. in terms of model theory, if we let α denote the assignment, $\langle \mathrm{Q}, < \rangle \models \phi[\alpha]$). We define a formula $\phi^\dagger$ in the language of the intervals over the rationals such that the assignment $[x_1 \leftarrow \langle l_1, r_1\rangle, x_2 \leftarrow \langle l_2, r_2\rangle, ..., x_3 \leftarrow \langle l_3, r_3\rangle]$ satisfies $\phi^\dagger$ in INT. So in terms of model theory, if β is this assignment, we are saying that $INT(Q) \models \phi^\dagger[\beta]$. The translation $\{\}^\dagger$ has this property for all first-order formulae in the respective languages, as was shown in *[Lad88.1, Lad87.5]*, but we shall only need it over a boolean subset of the language here.

We define the translation $\{\}^\dagger$. Any atomic formula of the language of RAT is of the form $x < y$ or $x = y$, and we shall consider correspondences for all possible assertions involving a primitive relation and rationals i_L, i_R, j_L, j_R that are endpoints of intervals i, j. We shall use the notation $i(R_1 + R_2 + ... + R_p)j$ to assert that the interval i is in one of the relations R_q to j (equivalent to $(R_1(i,j) \ \vee \ R_2(i,j) \ \vee \ ..... \vee \ R_p(i,j)))$.

It is easy to check the truth of the following statements connecting atomic truths about elements of RAT with disjunctions of atomic truths about elements of INT:

- $i_L < j_L \ \Leftrightarrow \ i\,(P + M + O + F^{\smile} + D^{\smile})\,j$
- $i_R < j_R \ \Leftrightarrow \ i\,(P + M + O + S + D)\,j$
- $i_L < j_R \ \Leftrightarrow \ i\,(P + M + O + Id(L) + S + F + D)\,j$
- $i_R < j_L \ \Leftrightarrow \ i\,P\,j$
- $i_L = j_L \ \Leftrightarrow \ i\,(S + Id(L) + S^{\smile})\,j$

- $i_R = j_R \;\Leftrightarrow\; i\,(F\;+\;Id(L)\;+\;F^{\smile})\,j$
- $i_L = j_R \;\Leftrightarrow\; i\,M^{\smile}\,j$
- $i_R = j_L \;\Leftrightarrow\; i\,M\,j$

It is also easy to convert these equivalences into assertions about formulas and satisfaction relations. First, we define a translation $-^{\dagger}$ from atomic formulas ϕ in the language of RAT into formulas $\phi^{\dagger}$ in the language of INT.

We use the small roman letters, some with superscripts,

$$p, d, o, m, s, f, p^{\smile}, d^{\smile}, o^{\smile}, m^{\smile}, s^{\smile}, f^{\smile}$$

to denote the twelve primitive predicate symbols in the language of INT.

We shall assume, in the list below, that $m \neq n$, since the cases where $m = n$ are either simply true or false (e.g. $f_n < g_n$ is true and $f_n = g_n$ false) under any assignment. The list of corresponding formulas ϕ and ψ in the languages of RAT and INT respectively are:

- If $\phi = (f_m < f_n)$ then
  $\phi^{\dagger} = (p(e_m, e_n)\;\vee\;m(e_m, e_n)\;\vee\;o(e_m, e_n)\;\vee\;f^{\smile}(e_m, e_n)\;\vee\;d^{\smile}(e_m, e_n))$
- If $\phi = (g_m < g_n)$ then
  $\phi^{\dagger} = (p(e_m, e_n)\;\vee\;m(e_m, e_n)\;\vee\;o(e_m, e_n)\;\vee\;s(e_m, e_n)\;\vee\;d(e_m, e_n))$
- If $\phi = (f_m < g_n)$ then
  $\phi^{\dagger} = (p(e_m, e_n) \vee m(e_m, e_n) \vee o(e_m, e_n) \vee s(e_m, e_n) \vee e_m = e_n \vee f(e_m, e_n) \vee d(e_m, e_n))$
- If $\phi = (g_m < f_n)$ then $\phi^{\dagger} = p(e_m, e_n)$
- If $\phi = (f_m = f_n)$ then
  $\phi^{\dagger} = (s(e_m, e_n)\;\vee\;e_m = e_n\;\vee\;s^{\smile}(e_m, e_n))$
- If $\phi = (g_m = g_n)$ then
  $\phi^{\dagger} = (f(e_m, e_n)\;\vee\;e_m = e_n\;\vee\;f^{\smile}(e_m, e_n))$
- If $\phi = (f_m = g_n)$ then
  $\phi^{\dagger} = m^{\smile}(e_m, e_n)$
- If $\phi = (g_m = f_n)$ then
  $\phi^{\dagger} = m(e_m, e_n)$

Now, $i_L < j_R$ is true in RAT iff $RAT \models (f_m < g_n)\{f_m \leftarrow i_L, g_n \leftarrow j_R\}$, and *mutatis mutandis* for the other atomic formulae. Similarly, in $INT(Q)$, $i(R_1 + R_2 + \ldots + R_p)j$ iff

$$INT \models (R_1(e_m, e_n) \ \vee \ldots \vee \ R_p(e_m, e_n))\{e_m \leftarrow i, e_n \leftarrow j\}$$

So, for example,

$$RAT \models (g_m < g_n)\{g_m \leftarrow i_R, g_n \leftarrow j_R\}$$

$$\Leftrightarrow$$

$$i_R < j_R$$

$$\Leftrightarrow$$

$$i\,(P \ + \ M \ + \ O \ + \ S \ + \ D)\,j$$

$$\Leftrightarrow$$

$$INT \models (p(e_m, e_n) \vee m(e_m, e_n) \vee o(e_m, e_n) \vee s(e_m, e_n) \vee d(e_m, e_n))\{e_m \leftarrow i, e_n \leftarrow j\}$$

$$\Leftrightarrow$$

$$INT \models \phi^\dagger\{e_m \leftarrow i, e_n \leftarrow j\}$$

and similarly for the other five types of atomic formulae in the language of RAT.

Hence, providing that the assignments of i_L, i_R, j_L, and j_R to f_m, g_m, f_n, and g_n, and i, j to e_m, e_n are made in accordance with the convention (about the intended correspondence of f_n, g_n, e_n), we may state a general lemma about the mutual satisfiability of ϕ and $\phi^\dagger$, whose proof follows immediately from the preceding considerations. We call such a pair of assignments *mutually acceptable.* If α is a mutually acceptable pair of assigments, let α_{RAT} be the part of α in the language of RAT, and α_{INT} be the corresponding part in the language of INT.

Lemma 3 *For every atomic formula ϕ of the language of RAT, for every mutually acceptable pair of assignments α,*

$$RAT \models \phi\{\alpha_{RAT}\} \ \Leftrightarrow \ INT \models \phi^\dagger\{\alpha_{INT}\}$$

Now we have defined the translation $\phi^\dagger$ of an atomic formula ϕ of the language of RAT, we extend this translation to all of the Boolean formulae of the language of RAT by the following clauses, which complete the inductive definition:

- If $\phi = (\neg\psi)$ then $\phi^\dagger = (\neg\psi^\dagger)$.
- If $\phi = (\psi \wedge \rho)$ then $\phi^\dagger = (\psi^\dagger \wedge \rho^\dagger)$.
- If $\phi = (\psi \vee \rho)$ then $\phi^\dagger = (\psi^\dagger \vee \rho^\dagger)$.

- If $\phi = (\psi \to \rho)$ then $\phi^\dagger = (\psi^\dagger \to \rho^\dagger)$.

We have the following extension of the lemma above:

Lemma 4 *For every Boolean formula ϕ of the language of RAT, for every mutually acceptable pair of assignments α,*

$$RAT \models \phi\{\alpha_{RAT}\} \;\;\Leftrightarrow\;\; INT \models \phi^\dagger\{\alpha_{INT}\}$$

5.3 From Interval Constraints to Point Constraints

We provide a syntactic translation from formulae in the language of INT to formulae in the language of RAT, that preserves satisfaction in the same sense as before. Namely, consider a formula ϕ in the language of INT in the variables $x_1, ..., x_n$ and suppose the assignment $[x_1 \;\leftarrow\; \langle l_1, r_1\rangle, x_2 \;\leftarrow\; \langle l_2, r_2\rangle, ...x_3 \;\leftarrow\; \langle l_3, r_3\rangle]$ satisfies ϕ (i.e. $INT \models \phi$ under the assigment above). We define a formula ϕ^* in the language of RAT with free variables $f_1, g_1, .., f_n, g_n$, such that the assignment $[f_1 \leftarrow l_1, g_1 \leftarrow r_1, f_2 \leftarrow l_2, g_2 \leftarrow r_2, ...f_n \leftarrow l_1, g_n \leftarrow r_1]$ satisfies the formula ϕ^* over RAT. So in terms of model theory, we are saying that $RAT \models \phi^*$ under the assignment above.

If ϕ is an atomic formula of the form $R(z, w)$, we define ϕ^* to be the formula $\phi_R(x, y, x', y')$, where if $z = e_n$, then $x = f_n$ and $y = g_n$, and similarly if $w = e_m$, then $x' = f_m$ and $y' = g_m$ (we need not worry about variable clashes since all of the formulas $\phi_R(x, y, x', y')$ are quantifier-free).

It was shown in *[Lad87.5]* that if $INT \models \phi$ where $\phi = R(z, w)$ then $RAT \models \phi^*$ as defined above, and vice versa. We can extend this property to Boolean combinations and quantifiers inductively. We may wish to do this since solving the $AGAN$ may not lead to an atomic formula on each edge, but in general may lead to an arbitrary Boolean formula. The preservation condition on satisfiability actually holds true for a translation of all first-order formulas, as shown in *[Lad87.5]*, and we give the full translation here for completeness.

- If $\phi = (\neg\psi)$ then $\phi^* = (\neg\psi^*)$.
- If $\phi = (\psi \wedge \rho)$ then $\phi^* = (\psi^* \wedge \rho^*)$.
- If $\phi = (\psi \vee \rho)$ then $\phi^* = (\psi^* \vee \rho^*)$.
- If $\phi = (\psi \to \rho)$ then $\phi^* = (\psi^* \to \rho^*)$.
- If $\phi = (\forall z)\psi$ and $z = e_n$ then $\phi^* = (\forall x \forall y)(x < y \;\to\; \psi^*)$ where $x = f_n$ and $y = g_n$

- If $\phi = \exists z \psi$ and $z = e_n$ then $\phi^* = (\exists x \exists y)(x < y \ \wedge \ \psi^*)$ where $x = f_n$ and $y = g_n$

The Boolean clauses just pass the translation through the Boolean connectives. The idea of the translation is that, if an interval satisfies ψ then its endpoints satisfy ψ^*. The choice of variables is bookkeeping.

Lemma 5 *Let ϕ be a formula in the language of INT. Then*

$$INT \models \phi\{e_1 \leftarrow i_1, \ldots\ldots, e_n \leftarrow i_n\}$$

iff

$$RAT \models \phi^*\{f_1 \leftarrow (i_1)_L, \ldots\ldots, f_n \leftarrow (i_n)_L, g_1 \leftarrow (i_1)_R, \ldots\ldots, g_n \leftarrow (i_n)_R\}$$

The proof of the lemma is to be found in *[Lad87.5]*.

We now consider in more detail certain special cases of $MN(A)$s and $AN(A)$s in which the information-passing techniques may be accomplished in feasible time.

6 Feasible Approximations

In this section, we discuss special cases of the general strategy for solving convex MINs that may be feasibly solved or approximated. In particular, we consider the special case where the $AN(A)$ is *pointisable [LadMad88.2]*.

6.1 From $MN(A)$ to $AN(A)$

The ideal information to pass from the metric constraints to the qualitative constraints is the information contained in $QC(A)$. $QC(A)$ was defined as a formula in *rational order normal form* RONF *[Lad87.5]*, as a disjunction of orderings of the variables f_i, g_i.

One might ask whether this is the most succinct form in which to pass information to the AN. The answer is no, and to find the most succinct form is precisely to compute the minimal disjunctive form that is equivalen to a given form, and we conjecture that this is NP-hard. Another possibility is that some approximation to the minimal form might be easily computable, and in this case it would be preferable to pass this information on to the AN. Exponential time might still be required to compute the $QC(A)$ form from this, so we do not advocate the computation of $QC(A)$ in general as the means to pass the qualitative information from $MN(A)$ to $AN(A)$. So although the $QC(A)$ form might not be the appropriate way to pass information to the AN, the information that needs to be passed is still logically equivalent to the $QC(A)$.

The technique we used earlier in the example computes the $WQC(A)$. $WQC(A)$ is computable in linear time in the number of edges, and therefore is a reasonable candidate for heuristic approximation to $QC(A)$.

6.2 Pointisable ANs

Pointisable qualitative networks are those in which the network corresponds directly to a single point algebra network with twice the number of nodes *[LadMad88.2]*. It was shown that there are 187 qualitative interval constraints which are pointisable, and hence any network in which each label is one of these is a pointisable network, and *vice versa*. The importance of pointisable networks is that there is an $O(n^2)$ satisfaction algorithm for path-consistent networks, giving an $O(n^3)$ algorithm for satisfiability *op. cit.*.

If the translation is performed, of a pointisable AN to its qualitative point algebra equivalent (a linear time operation in the number of edges of the network), then there is no need to perform the translations of the previous section between the quantitative and the qualitative parts of the constraint network. The translated network may employ the same variables f_i, g_i that have been used to define the quantitative constraints. The translation is linear in the number of nodes and edges of the original interval network *[op. cit.]*, and the labels on an edge are from the point algebra, consisting only of $<, >, =, \leq, \geq, \neq, \emptyset$ and 1, into which the constraints from the $QC(A)$, or its approximations, may be incorporated directly. Similarly, there is no need to translate solutions to the qualitative network back into the vocabulary of the quantitative network, to apply the earlier technique of using the point orderings to prune the ranges, since the vocabulary is substantially the same.

In the case where we may expect the solution of the $AGAN$ to involve pointisable formulae, i.e. the solution consists of a reduction of the $AGAN$ in which all labels are pointisable relations, then the Boolean equivalents to the formulae are contained in the appendix to *[Lad88.2]*, and are also reprinted in *[vBeCoh89]*. There are 187 of them, and we do not reproduce them here. The pointisable relations are all the relations between two intervals that may be represented as a conjunction of atomic relations on the endpoints, and thus they are a suitable target for this stage of the solution process. (Basically disjunctions either lead to exponential growth of potential solution networks, in the case of exhaustive search, or to a reduction in the information being passed on to the metric network. We observed this phenomenon with the QC and WQC constraints, in the case of the translation the other way).

We should note also that an AN that is not pointisable as stated may become so when information from the qualitative network is incorporated, or when reduction techniques are applied, so the techniques mentioned for dealing with pointisable networks may be applicable at any stage as a qualitative network becomes pointisable.

6.3 All-Solutions Solutions

In the case in which it is desired to compute all solutions to a given constraint network, then some of the above techniques become harder. The minimal network corresponding to a convex metric network is still easily computable, but the minimal network corresponding to even a point algebra network (the case of a pointisable AN) may not be (as investigated in *[vBeCoh89]*). Much further work needs to be done in this area.

7 Practical Use of the Approach

We have advocated the use of a time unit system, the TUS, for reasoning about real intervals of time *[Lad87.2, LadMad89.1]*. The system represents units of time as a sequence of nested units in the same manner as does a clock or calendar. We showed in *[op. cit.]* that the TUS was indeed a countable model of the interval algebra and thus is isomorphic as a structure to the pairs-of-rationals interpretation of intervals (since the interval algebra is countably categorical). It would be unfortunate if our approach needed to make use of substantial translations (e.g. as provided in *[op. cit.]*) in order to apply to the TUS. This is not the case, as we indicate here. Lengths may be measured in the TUS as a multiple of units, for example 3 *days*, 5 *years*, 200 *minutes*. So the first modification to note is that all measures include units which are *basic units* in the TUS. So every range has its accompanying unit, and when calculations are performed amongst ranges, as in computing the minimal distance network, account must be taken to ensure that the units are converted into common units before a calculation is made. Since every constraint network is finite, there is indeed a 'least common unit' for the range constraints (since all the units are nested, it is the case that one is always an integral multiple of any other or *vice versa*). We don't advocate any specific method of maintaining these units denominators since it seems to us to be just easy bookkeeping to do so.

Many of the metric constraints may take a different, more suitable, form in the TUS. Metric constraints of the form $g_i - f_i \in R$, for example, refer to the *duration* of an interval in terms of the units in which R is expressed. The TUS also includes the notion of the *difference* of two intervals (the length of the interval between them, expressed in a suitable unit), and the *overlap size* of intervals. Using these notions it is an easy matter to express the other three types of constraints (i.e. $f_i - f_j$, $f_i - g_j$, $g_i - f_j$). Hence the vocabulary of the TUS provides the hooks for implementing

DMP-type metric constraints directly, without the need to refer to points. For those who wish to discuss the relative merits of point or interval formulations of temporal reasoning, it may be useful to note that the discussion above shows that the DMP approach to metric constraints does not force us to use a point-based formalisation (as might have been thought). The TUS is purely an interval formulation, and the metric constraints work just as well in the manner indicated above as they do with points.

8 Further Research

The most obvious open questions arising from this work have to do with the $QC(A)$ and its equivalents. What is the complexity of relations between the $QC(A)$ and its minimal equivalents (which we'd prefer to compute). Are any of these equivalent formulas feasibly computable from an arbitrary minimal distance graph (the minimal representation of the metric network)? And finally, we have not provided a completeness proof for our suggested procedure using the translation from metric to $QC(A)$, solving the $AGAN$, and passing the information back in the manner we suggested to the metric network. There are many aspects to such a proof. For example, it should be clear that the $QC(A)$ is the information needed to add to the AN - but we have not provided a proof. Similarly, suppose one were to pass the $QC(A)$ information to the AN, solve it (say, find the minimal network) and pass all the possible orderings of endpoints (via the translation $\{\}^*$) back to the metric network in the form of prunings, do we then obtain a minimal network for the whole problem (i.e. a network which contains only values for the nodes, and the differences, which are extensible into a solution of the complete network, including the qualitative constraints, and such that every such solution is contained in some assigment of values from the nodes)? We have shown here only that a reduction of the minimal network is computed by this technique.

On the implementation front, it also remains to incorporate these metric constraint solution techniques into the TUS using the approach suggested in the previous section.

9 Summary

We have shown how the metric constraint satisfaction techniques of Dechter, Meiri and Pearl may be incorporated into an interval constraint satisfaction problem which includes a qualitative component. Our approach envisions a constraint network in which both kinds of constraints appear between two nodes. The nodes are to be interpreted by intervals, and the metric information between two nodes is captured by a set of up to four metric constraints between four variables interpreted as the endpoints of the nodes. We characterised the information that needed to be passed from the metric constraints to augment the qualitative constraints, and although we conjectured that the complete information in relevant form might be NP-hard to compute, we suggested a suitable approximation that may be computed in linear time in the number of edges. We showed how solutions from the qualitative component of the network may be passed back to the metric component to compute a reduction of the minimal network (and therefore a sound, if not complete, solution).

We noted that there is no real reliance on points in this approach, since the endpoint constraints may all be phrased in terms of operations on intervals in the time unit system TUS, which has only proper intervals, and no points, in it.

We concluded with a list of problems for future research.

Bibliography

All83 : Allen, J.F., *Maintaining Knowledge about Temporal Intervals*, Comm. A.C.M. 26 (11), November 1983, 832-843.

AllHay88 : Allen J.F. and Hayes, P. J., *Moments and Points in an Interval-Based Temporal Logic*, Technical Report TR180, Dept. of Computer Science, University of Rochester, December 1987.

ChaKei73 : Chang, C.C., and Keisler, H.J., *Model Theory*, North-Holland, 1973.

DecPea88 : Dechter, R., and Pearl, J., *Network-Based Heuristics for Constraint-Satisfaction Problems*, Artificial Intelligence 34, 1988, 1-38.

DeMePe89 : Dechter, R., Meiri, I., and Pearl, J., *Temporal Constraint Satisfaction*, Proceedings of the First International Conference on Knowledge Representation and Reasoning (KR89), Morgan Kaufmann, 1989, pp 83-93.

Fre78 : Freuder, E.C., *Synthesizing Constraint Expressions*, Communications of the ACM 21 (11), Nov 1978, 958-966.

Güs89 : Güsgen, H.W., *A Universal Constraint Programming Language*, Proceedings of the 11th IJCAI, 1989, Detroit, Michigan.

GüsHer88 : Güsgen, H.W., and Hertzberg, J., *Properties of Local Constraint Propagation*, Artificial Intelligence 36, 1988, 237-247.

HeMoTa?? : Henkin, L., Monk, D., Tarski, A., Andréka, H., and Németi, I., *Cylindric Set Algebra*, Lecture Notes in Mathematics, , Springer, 19??.

Lad87.2 : Ladkin, P.B., *The Completeness of a Natural System for Reasoning with Time Intervals*, Proceedings of the 10th International Joint Conference on Artificial Intelligence (Milano, Italy), Morgan Kaufmann 1987, 462-467, also available as Kestrel Institute Technical Report KES.U.87.5.

Lad87.3 : Ladkin, P.B., *Models of Axioms for Time Intervals*, Proceedings of AAAI-87, the Sixth National Conference on Artificial Intelligence, Morgan Kaufmann 1987, 234-239, also available in a longer version as Kestrel Institute Technical Report KES.U.87.4.

Lad87.5 : Ladkin, P.B., *Constraint Satisfaction in Time Interval Structures I: Convex Intervals*, Kestrel Institute Technical Report KES.U.87.11, 1987.

Lad88.1 : Ladkin, P.B., *Satisfying First-Order Constraints about Time Intervals*, Proceedings of the 7th National Conference on AI, (AAAI-88), 512-517, Morgan Kaufmann 1988.

LadMad87 : Ladkin, P.B. and Maddux, R.D., *The Algebra of Convex Time Intervals*, Kestrel Institute Technical Report KES.U.87.2.

LadMad88.1 : Ladkin, P.B., and Maddux, R.D., *Representation and Reasoning with Convex Time Intervals*, Kestrel Institute Technical Report KES.U.88.2, also submitted for publication.

LadMad88.2 : Ladkin, P.B., and Maddux, R.D., *On Binary Constraint Networks*, Kestrel Institute Technical Report KES.U.88.8.

LadMad88.3 : Ladkin, P.B., and Maddux, R.D., *On the Parallel Complexity of Path-Consistency Algorithms*, preprint.

LadMad89.1 : Ladkin, P.B., and Maddux, R.D., *The Algebra of Constraint Satisfaction and Temporal Reasoning*, submitted for publication.

Mac77 : Mackworth, A.K., *Consistency in Networks of Relations*, Artificial Intelligence 8, 1977, 99-118.

Mac87 : Mackworth, A.K., *Constraint Satisfaction*, in the *Encyclopedia of Artificial Intelligence*, ed. S. Shapiro, Wiley Interscience 1987.

MacFre85 : Mackworth, A.K., and Freuder, E.C., *The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems*, Artificial Intelligence 25, 65-74, 1985.

Mei89 : Meiri, I., and others, *post-presentation discussion of DMP at KR89 conference.*

MohHen86 : Mohr, R., and Henderson, T.C., *Arc and Path Consistency Revisited*, Artificial Intelligence 28, 1986, 225-233.

Mon74 : Montanari, U., *Networks of Constraints: Fundamental Properties and Applications to Picture Processing*, Inform. Sci. 7, 1974, 727-732.

Smi88 : Smith, D.R., *The Structure and Design of Global Search Algorithms*, Kestrel Institute Technical Report KES.U.87.12, 1988; also submitted for publication.

Tar41 : Tarski, A., *On the Calculus of Relations*, Journal of Symbolic Logic 6, 1941, 73-89.

vBeCoh89 : van Beek, P., and Cohen, R., *Approximation Algorithms for Temporal Reasoning*, Proceedings of the 11th IJCAI, 1989, Detroit, Michigan.

VilKau86 : Vilain, M., and Kautz, H., *Constraint Propagation Algorithms for Temporal Reasoning*, Proceedings of AAAI-86, 377-382, Morgan Kaufmann, 1986.