Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

COMBINING QUALITATIVE AND QUANTITATIVE CONSTRAINTS IN TEMPORAL REASONING

Itay Meiri

July 1991
CSD-910030

To appear in Proceedings, AAAI-91, Anaheim, CA., July 1991.

TECHNICAL REPORT
R-160
April, 1991

Combining Qualitative and Quantitative Constraints in Temporal Reasoning*

Itay Meiri
Cognitive Systems Laboratory
Computer Science Department
University of California, Los Angeles, CA 90024
itay@cs.ucla.edu

Abstract

This paper presents a general model for temporal reasoning, capable of handling both qualitative and quantitative information. This model allows the representation and processing of all types of constraints considered in the literature so far, including metric constraints (restricting the distance between time points), and qualitative, disjunctive, constraints (specifying the relative position between temporal objects). Reasoning tasks in this unified framework are formulated as constraint satisfaction problems, and are solved by traditional constraint satisfaction techniques, such as backtracking and path consistency. A new class of tractable problems is characterized, involving qualitative networks augmented by quantitative domain constraints, some of which can be solved in polynomial time using arc and path consistency.

1 Introduction

In recent years, several constraint-based formalisms have been proposed for temporal reasoning, most notably, Allen's interval algebra (IA) [1], Vilain and Kautz's point algebra (PA) [14], Dean and McDermott's time map [2], and metric networks (Dechter, Meiri and Pearl [4]). In these formalisms, temporal reasoning tasks are formulated as constraint satisfaction problems, where the variables are temporal objects such as points and intervals, and temporal statements are viewed as constraints on the location of these objects along the time line. Unfortunately, none of the existing formalisms can conveniently handle all forms of temporal knowledge. Qualitative approaches such as Allen's interval algebra and Vilain and Kautz's point algebra face difficulties in representing and reasoning about metric, numerical information, while the quantitative approaches exhibit limited expressiveness when it comes to qualitative information [4].

*This work was supported in part by the Air Force Office of Scientific Research, AFOSR 900136.

In this paper we offer a general, network-based computational model for temporal reasoning, capable of handling both qualitative and quantitative information. In this model, variables represent both points and intervals (as opposed to existing formalisms, where one has to commit to a single type of objects), and constraints may be either metric, between points, or qualitative disjunctive relations between objects. The unique feature of this framework is that it allows the representation and processing of all types of constraints considered in the literature so far.

The main contribution of this paper lies in providing a formal unifying framework for temporal reasoning, generalizing the interval algebra, the point algebra, and metric networks. In this framework, we are able to utilize constraint satisfaction techniques in solving several reasoning tasks. Specifically:

1. General networks can be solved by decomposition into *singleton labelings*, each solvable in polynomial time. This decomposition scheme can be improved by traditional constraint satisfaction techniques such as variants of backtrack search.
2. The input can be effectively encoded in a *minimal network* representation, which provides answers to many queries.
3. Path consistency algorithms can be used in preprocessing the input network to improve search efficiency, or to compute an approximation to the minimal network.
4. We were able to identify two classes of tractable problems, solvable in polynomial time. The first consists of *augmented qualitative networks*, composed of qualitative constraints between points and quantitative domain constraints, which can be solved using arc and path consistency. The second class consists of networks for which path consistency algorithms are exact.

We also show that our model compares favorably with an alternative approach for combining quantitative and qualitative constraints, proposed by Kautz and

Ladkin [6], from both conceptual and computational points of view.

The paper is organized as follows. Section 2 formally defines the constraint types under consideration. The definitions of the new model are given in Section 3. Section 4 reviews and extends the hierarchy of qualitative networks. Section 5 discusses *augmented qualitative networks*—qualitative networks augmented by domain constraints. Section 6 presents two methods for solving general networks: a decomposition scheme and path consistency, and identifies a class of networks for which path consistency is exact. Section 7 provides summary and concluding remarks, including a comparison to Kautz and Ladkin's model. Proofs of theorems can be found in the extended version of this paper [10].

2 The Representation Language

Consider a typical temporal reasoning problem. We are given the following information.

Example 1. John and Fred work for a company in LA. They usually work at the local office, in which case it takes John less than 20 minutes and Fred between 15–20 minutes to get to work. Twice a week John works at the main office, in which case he commutes at least 60 minutes to work. Today John left home between 7:05–7:10, and Fred arrived at work between 7:50–7:55. We also know that Fred and John met at a traffic light on their way to work.

We wish to represent and reason about such knowledge. We wish to answer queries such as: "is the information in this story consistent?," "who was the first to arrive at work?," "what are the possible times at which John arrived at work?," and so on.

We consider two types of temporal objects: points and intervals. Intervals correspond to time periods during which events occur or propositions hold, and points represent beginning and ending points of some events, as well as neutral points of time. For example, in our story, we have two meaningful events: "John was going to work" and "Fred was going to work." These events are associated with intervals $J = [P_1, P_2]$, and $F = [P_3, P_4]$, respectively. The extreme points of these intervals, $P_1, \ldots, P_4$, represent the times in which Fred and John left home and arrived at work. We also introduce a neutral point, P_0, to represent the "beginning of the world" in our story. One possible choice for P_0 is 7:00 a.m. Temporal statements in the story are treated as constraints on the location of objects (such as intervals J and F, and points $P_0, \ldots, P_4$) along the time line. There are two types of constraints: qualitative and quantitative. Qualitative constraints specify the relative position of pairs of objects. For instance, the fact that John and Fred met at a traffic light, forces intervals J and F to overlap. Quantitative constraints place absolute bounds or restrict the temporal distance between points. For example, the information on Fred's commuting time constrains the length of interval F, i.e., the distance between P_3 and P_4. In the rest of this section we formally define qualitative and quantitative constraints, and the relationships between them.

Relation	Symbol	Inverse	Relations on Endpoints
p before I	*b*	*bi*	$p < I^-$
p starts I	*s*	*si*	$p = I^-$
p during I	*d*	*di*	$I^- < p < I^+$
p finishes I	*f*	*fi*	$p = I^+$
p after I	*a*	*ai*	$p > I^+$

Table 1: The basic relations between a point p and an Interval $I = [I^-, I^+]$.

Qualitative Constraints

A qualitative constraint between two objects O_i and O_j, each may be a point or an interval, is a disjunction of the form

$$(O_i\ r_1\ O_j) \vee \cdots \vee (O_i\ r_k\ O_j), \qquad (1)$$

where each one of the r_i's is a *basic relation* that may exist between the two objects. There are three types of basic relations.

- Basic *Interval-Interval (II) relations* that can hold between a pair of intervals [1], *before, meets, starts, during, finishes, overlaps*, their inverses, and the equality relation, a total of 13 relations, denoted by the set $\{b, m, s, d, f, o, bi, mi, si, di, fi, oi, =\}$.
- Basic *Point-Point (PP) relations* that can hold between a pair of points [14], denoted by the set $\{<, =, >\}$.
- Basic *Point-Interval (PI) relations* that can hold between a point and an interval, and basic *Interval-Point (IP) relations* that can hold between an interval and a point. These relations are defined in Table 1 (see also [7]).

A subset of basic relations (of the same type) corresponds to an ambiguous, disjunctive, relationship between objects. For example, Equation (1) may also be written as $O_i\ \{r_1, \ldots, r_k\}\ O_j$; alternatively, we say that the constraint between O_i and O_j is the relation set $\{r_1, \ldots, r_k\}$. One qualitative constraint given in Example 1 reflects the fact that John and Fred met at a traffic light. It is expressed by an II relation specifying that intervals J and F are not disjoint:

$$J\ \{s, si, d, di, f, fi, o, oi, =\}\ F.$$

To facilitate the processing of qualitative constraints, we define a *qualitative algebra(QA)*, whose elements are all legal constraints (all subsets of basic relations of the same type)—2^{13} II Relations, 2^3 PP relations, 2^5 PI relations, and 2^5 IP relations. Two binary operations are defined on these elements: intersection and composition. The *intersection* of two qualitative constraints.

	PP	PI	IP	II
PP	$[T_{PA}]$	$[T_1]$	$[\emptyset]$	$[\emptyset]$
PI	$[\emptyset]$	$[\emptyset]$	$[T_2]$	$[T_4]$
IP	$[T_1]^t$	$[T_3]$	$[\emptyset]$	$[\emptyset]$
II	$[\emptyset]$	$[\emptyset]$	$[T_4]^t$	$[T_{IA}]$

Table 2: A full transitivity table.

R' and R'', denoted by $R' \oplus R''$, is the set-theoretic intersection $R' \cap R''$. The *composition* of two constraints, R' between objects O_i and O_j, and R'' between objects O_j and O_k, is a new relation between objects O_i and O_k, induced by R' and R''. Formally, the composition of R' and R'', denoted by $R' \otimes R''$, is the composition of the constituent basic relations, namely

$$R' \otimes R'' = \{r' \otimes r'' | r' \in R', r'' \in R''\}.$$

Composition of two basic relations r' and r'', is defined by a *transitivity table* shown in Table 2. Six transitivity tables, $T_1, \ldots, T_4, T_{PA}, T_{IA}$, are required; each defining a composition of basic relations of a certain type. For example, composition of a basic PP relation and a basic PI relation is defined in table T_1. Two important subsets of QA are Allen's Interval Algebra (IA), the restriction of QA to II relations, and Vilain and Kautz's Point Algebra (PA), its restriction to PP relations. The corresponding transitivity tables are given in [1] and [14], and appear in Table 2 as T_{IA} and T_{PA}, respectively. The rest of the tables, $T_1, \ldots, T_4$, are given in the extended version of this paper [10]. Illegal combinations in Table 2 are denoted by $\emptyset$.

Quantitative Constraints

Quantitative constraints refer to absolute location or the distance between *points* [4]. There are two types of quantitative constraints:

- A *unary* constraint, on point P_i, restricts the location of P_i to a given set of intervals

$$(P_i \in I_1) \vee \cdots \vee (P_i \in I_k).$$

- A *binary* constraint, between points P_i and P_j, constrains the permissible values for the distance $P_j - P_i$:

$$(P_j - P_i \in I_1) \vee \cdots \vee (P_j - P_i \in I_k).$$

In both cases the constraint is represented by a set of intervals $\{I_1, \ldots, I_k\}$; each interval may be open or closed in either side[1]. For example, one binary constraint given in our story specifies the duration of interval J (the event "John was going to work)":

$$P_2 - P_1 \in \{(0, 20), (60, \infty)\}.$$

[1]The set $\{I_1, \ldots, I_k\}$ represents the set of real numbers $I_1 \cup \cdots \cup I_k$. Throughout the paper we shall use the convention whereby a real number v is in $\{I_1, \ldots, I_k\}$ iff $v \in I_1 \cup \cdots \cup I_k$.

The fact that John left home between 7:05–7:10 is translated into a unary constraint on P_1, $P_1 \in \{(5, 10)\}$, or $5 < P_1 < 10$ (note that all times are relative to P_0, i.e. 7:00 a.m.). Sometimes it is easier to treat a unary constraint on P_i as a binary constraint between P_0 and P_i, having the same interval representation. For example, the above unary constraint is equivalent to the binary constraint, $P_1 - P_0 \in \{(5, 10)\}$.

The intersection and composition operations for quantitative constraints assume the following form. Let C' and C'' be quantitative constraints, represented by interval sets I' and I'', respectively. Then, the intersection of C' and C'' is defined as

$$C' \oplus C'' = \{x | x \in I', x \in I''\}.$$

The composition of C' and C'' is defined as

$$C' \otimes C'' = \{z | \exists x \in I', \exists y \in I'', \ x + y = z\}.$$

Relationships between Qualitative and Quantitative Constraints

The existence of a constraint of one type sometimes implies the existence of an implicit constraint of the other type. This can only occur when the constraint involves two points. Consider a pair of points P_i and P_j. If a quantitative constraint, C, between P_i and P_j is given (by an interval set $\{I_1, \ldots, I_k\}$), then the implied qualitative constraint, $\mathrm{QUAL}(C)$, is defined as follows (see also [6]).

- If $0 \in \{I_1, \ldots, I_k\}$, then "=" $\in \mathrm{QUAL}(C)$.
- If there exists a value $v > 0$ such that $v \in \{I_1, \ldots, I_k\}$, then "<" $\in \mathrm{QUAL}(C)$.
- If there exists a value $v < 0$ such that $v \in \{I_1, \ldots, I_k\}$, then ">" $\in \mathrm{QUAL}(C)$.

Similarly, If a qualitative constraint, C, between P_i and P_j is given (by a relation set R), then the implied quantitative constraint, $\mathrm{QUAN}(C)$, is defined as follows.

- If "<" $\in R$, then $(0, \infty) \in \mathrm{QUAN}(C)$.
- If "=" $\in R$, then $[0] \in \mathrm{QUAN}(C)$.
- If ">" $\in R$, then $(-\infty, 0) \in \mathrm{QUAN}(C)$.

The intersection and composition operations can be extended to cases where the operands are constraints of different types. If C' is a quantitative constraint and C'' is qualitative, then intersection is defined as quantitative intersection:

$$C' \oplus C'' = C' \oplus \mathrm{QUAN}(C''). \qquad (2)$$

Composition, on the other hand, depends on the type of C''.

- If C'' is a PP relation, then composition (and consequently the resulting constraint) is quantitative

$$C' \otimes C'' = C' \otimes \mathrm{QUAN}(C'').$$

- If C'' is a PI relation, then composition is qualitative

$$C' \otimes C'' = \mathrm{QUAL}(C') \otimes C''.$$

3 General Temporal Constraint Networks

We now present a network-based model which facilitates the processing of all constraints described in the previous section. The definitions of the new model follow closely those developed for discrete constraint networks [11], and for metric networks [4].

A *general temporal constraint network* involves a set of variables $\{X_1, \ldots, X_n\}$, each representing a temporal object (a point or an interval), and a set of unary and binary constraints. When a variable represents a time point its domain is the set of real numbers $\Re$; when a variable represents a temporal interval, its domain is the set of ordered pairs of real numbers, i.e. $\{(a,b)|a,b \in \Re, a < b\}$. Constraints may be quantitative or qualitative. Each qualitative constraint is represented by a relation set R. Each quantitative constraint is represented by an interval set I. Constraints between variables representing points are always maintained in their quantitative form. We also assume that unary quantitative constraints are represented by equivalent binary constraints, as shown in the previous section. A set of internal constraints relates each interval $I = [I^-, I^+]$ to its endpoints, I^- $\{starts\}$ I, and I^+ $\{finishes\}$ I.

A constraint network is associated with a *directed constraint graph*, where nodes represent variables, and an arc $i \rightarrow j$ indicates that a constraint C_{ij}, between variables X_i and X_j, is specified. The arc is labeled by an interval set (when the constraint is quantitative) or by a QA element (when it is qualitative). The constraint graph of Example 1 is shown in Figure 1.

A tuple $X = (x_1, \ldots, x_n)$ is called a *solution* if the assignment $\{X_1 = x_1, \ldots, X_n = x_n\}$ satisfies all the constraints (note that the value assigned to a variable which represents an interval is a pair of real numbers). The network is *consistent* if at least one solution exists. A value v is a *feasible value* for variable X_i, if there exists a solution in which $X_i = v$. The set of all feasible values of a variable is called its *minimal domain*.

We define a partial order, $\subseteq$, among binary constraints of the *same type*. A constraint C' is *tighter* than constraint C'', denoted by $C' \subseteq C''$, if every pair of values allowed by C' is also allowed by C''. If C' and C'' are qualitative, represented by relation sets R' and R'', respectively, then $C' \subseteq C''$ if and only if $R' \subseteq R''$. If C' and C'' are quantitative, represented by interval sets I' and I'', respectively, then $C' \subseteq C''$ if and only if for every value $v \in I'$, we have also $v \in I''$. This partial order can be extended to networks in the usual way. A network N' is tighter than network N'', if the partial order $\subseteq$ is satisfied for all the corresponding constraints. Two networks are *equivalent* if they possess the same solution set. A network may have many equivalent representations; in particular, there is a unique equivalent network, M, which is minimal with respect to $\subseteq$, called the *minimal network* (the minimal network is unique because equivalent networks are closed under intersection). The arc constraints specified by M are called the *minimal constraints*.

The minimal network is an effective, more explicit, encoding of the given knowledge. Consider for example the minimal network of Example 1. The minimal constraint between J and F is $\{di\}$, and the minimal constraint between P_1 and P_2 is $\{(60, \infty)\}$. From this minimal network representation, we can infer that today John was working in the main office; he arrived at work after 8:00 a.m., and thus Fred was the first to arrive at work.

Given a network N, the first interesting task is to determine its consistency. If the network is consistent, we are interested in other reasoning tasks, such as finding a solution to N, computing the minimal domain of a given variable X_i, computing the minimal constraint between a given pair of variables X_i and X_j, and computing the full minimal network. The rest of the paper is concerned with solving these tasks.

4 The Hierarchy of Qualitative Networks

Before we present solution techniques for general networks, we briefly describe the hierarchy of qualitative networks.

Consider a network having only qualitative constraints. If all constraints are II relations (namely IA elements), or PP relations (PA elements), then the network is called an *IA network*, or a *PA network*, respectively [12]. If all constraints are PI and IP relations, then the network is called an *IPA network* (for *Interval-Point* Algebra[2]). A special case of a PA network, where the relations are convex (taken only from $\{<, \leq, =, \geq, >\}$, namely excluding $\neq$), is called a *convex PA network* (*CPA network*).

It can be easily shown that any qualitative network can be represented by an IA network. On the other hand, there are some qualitative networks that cannot be represented by a PA network. For example (see [14]), a network consisting of two intervals, I and J, and a single constraint between them, I $\{before, after\}$ J. Formally, the following relationship can be established among qualitative networks.

Proposition 1 *Let QN be the set of all qualitative networks. Let $net(CPA)$, $net(PA)$, $net(IPA)$, and $net(IA)$ denote the set of qualitative networks which can be represented by CPA networks, PA networks, IPA networks, and IA networks, respectively. Then,*

$$net(CPA) \subset net(PA) \subset net(IPA) \subset net(IA) = QN.$$

[2]We use this name to comply with the names IA and PA, although technically these relations, together with the intersection and composition operations, do not constitute an algebra, because they are not closed under composition.

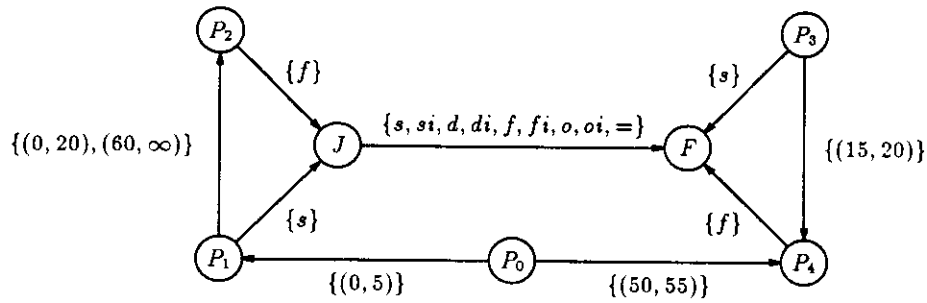

Figure 1: The constraint graph of Example 1.

By climbing up in this hierarchy from CPA networks towards IA networks we gain expressiveness, but at the same time lose tractability. For example, deciding consistency of a PA network can be done in time $O(n^2)$ [13], but it becomes NP-complete for IA networks [14], or even for IPA networks, as stated in the following theorem.

Theorem 2 *Deciding consistency of an IPA network is NP-hard.*

Theorem 2 suggests that the border between tractable and intractable qualitative networks lies somewhere between PA networks and IPA networks.

5 Augmented Qualitative Networks

We now return to solving general networks. First, we observe that even the simplest task of deciding consistency of a general network is NP-hard. This follows trivially from the fact that deciding consistency for either metric networks or IA networks is NP-hard [4, 14]. Therefore, it is unlikely that there exists a general polynomial algorithm for deciding consistency of a network. In this section we take another approach, and pursue "islands of tractability"—special classes of networks which admit polynomial solution. In particular, we consider the simplest type of network which contains both qualitative and quantitative constraints, called an *augmented qualitative network*, a qualitative network augmented by unray constraints on its domains.

We may view qualitative networks as a special case of augmented qualitative networks, where the domains are unlimited. For example, PA networks can be regarded as augmented qualitative networks with domains $(-\infty, \infty)$. It follows that in our quest for tractability, we can only augment tractable qualitative networks such as CPA and PA networks.

In this section, we consider CPA and PA networks over three domain classes which carry significant importance in temporal reasoning applications:

1. *Discrete domains*, where each variable may assume only a finite number of values (for instance, when we settle for crude timing of events such as the day or year in which they occurred).
2. *Single-interval domains*, where we have only an upper and/or a lower bound on the timing of events.
3. *Multiple-intervals domains*, which subsumes the two previous cases[3].

A CPA network over multiple-intervals domains is depicted in Figure 2a, where each variable is labeled by its domain intervals. Note that in this example, and also throughout the rest of this section, we use a special constraint graph representation, where the domain constraints are expressed as *unary* constraints (in general networks they are represented as binary constraints).

We next show that for augmented CPA networks and for some augmented PA networks, all interesting reasoning tasks can be solved in polynomial time, by enforcing arc consistency (AC) and path consistency (PC).

First, let us review the definitions of arc consistency and path consistency [8, 11]. An arc $i \rightarrow j$ is *arc consistent* if and only if for any value $x \in D_i$, there is a value $y \in D_j$, such that the constraint C_{ij} is satisfied. A path P from i to j, $i_0 = i \rightarrow i_1 \rightarrow \cdots \rightarrow i_m = j$, is *path consistent* if the direct constraint C_{ij} is tighter than the composition of the constraints along P, namely, $C_{ij} \subseteq C_{i_0,i_1} \otimes \cdots \otimes C_{i_{m-1},i_m}$. Note that our definition of path consistency is slightly different than the original one [8], as it does not consider the domain constraints. A *network* G is arc (path) consistent if all its arcs (paths) are consistent. Figure 2b shows an equivalent arc- and path-consistent form of the network in Figure 2a.

The following theorems establish the local consistency levels which are sufficient to determine the consistency of augmented CPA networks.

Theorem 3 *Any nonempty[4] arc-consistent CPA network over discrete domains is consistent.*

Theorem 4 *Any nonempty arc- and path-consistent CPA network over multiple-intervals domains is consistent.*

[3] Note that a discrete domain $\{v_1, \ldots, v_k\}$ is essentially a multiple-intervals domain $\{[v_1, v_1], \ldots, [v_k, v_k]\}$.

[4] A nonempty network is a network in which all domains and all constraints are nonempty.

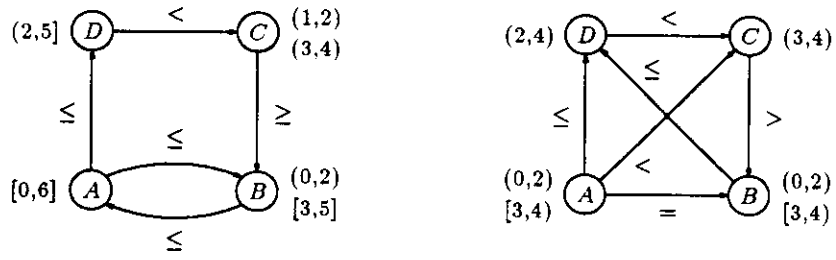

Figure 2: (a) A CPA network over multiple-intervals domains. (b) An equivalent arc- and path-consistent form.

Theorems 3 and 4 provide an effective test for deciding consistency of an augmented CPA network. We simply enforce the required consistency level, and then check whether the domains and the constraints are empty. The network is consistent if and only if all domains and all constraints are nonempty. Moreover, by enforcing arc and path consistency we also compute the minimal domains, as stated in the next theorem.

Theorem 5 *Let $G = (V, E)$ be a nonempty arc- and path-consistent CPA network over multiple-intervals domains. Then, all domains are minimal.*

When we move up in the hierarchy from CPA networks to PA networks (allowing also the inequality relation between points), deciding consistency and computing the minimal domains remain tractable for single-interval domains. Unfortunately, deciding consistency becomes NP-hard for discrete domains, and consequently, for multiple-intervals domains.

Theorem 6 *Let $G = (V, E)$ be a nonempty arc- and path-consistent PA network over single-interval domains. Then, G is consistent, and all domains are minimal.*

Proposition 7 *Deciding consistency of a PA network over discrete domains is NP-hard.*

One way to convert a network into an equivalent arc-consistent form is by applying algorithm AC-3 [8], shown in Figure 3. The algorithm repeatedly applies the function REVISE$((i, j))$, which makes arc $i \rightarrow j$ consistent, until a fixed point, where all arcs are consistent, is reached. The function REVISE$((i, j))$ restricts D_i, the domain of variable X_i, using operations on quantitative constraints:

$$D_i \leftarrow D_i \oplus D_j \otimes \mathrm{QUAN}(C_{ji}).$$

Taking advantage of the special features of PA networks, we are able to bound the running time of AC-3 as follows.

Theorem 8 *Let $G = (V, E)$ be an augmented PA network. Let n and e be the number of nodes and the number of edges, respectively. Then, the timing of algorithm AC-3 is bounded as follows.*

1. $Q \leftarrow \{i \rightarrow j | i \rightarrow j \in E\}$
2. **while** $Q \neq \emptyset$ **do**
3. select and delete any arc $k \rightarrow m$ from Q
4. **if** REVISE$((k, m))$ **then**
5. $Q \leftarrow Q \cup \{i \rightarrow k | i \rightarrow k \in E, i \neq m\}$
6. **end**

Figure 3: AC-3—an arc consistency algorithm.

- *If the domains are discrete, then AC-3 takes $O(ek \log k)$ time, where k is the maximum domain size*[5].
- *If the domains consist of single intervals, then AC-3 takes $O(en)$ time.*
- *If the domains consist of multiple intervals, then AC-3 takes $O(en^2K^2)$ time, where K is the maximum number of intervals in any domain.*

A network can be converted into an equivalent path-consistent form by applying any path consistency algorithm to the underlying qualitative network [8, 14, 12]. Path consistency algorithms impose local consistency among triplets of variables, (i, k, j), by using a relaxation operation

$$C_{ij} \leftarrow C_{ij} \oplus C_{ik} \otimes C_{kj}. \tag{3}$$

Relaxation operations are applied until a fixed point is reached, or until some constraint becomes empty indicating an inconsistent network. One efficient path consistency algorithm is PC-2 [8], shown in Figure 4, where the relaxation operation of Equation (3) is performed by the function REVISE$((i, k, j))$. Algorithm PC-2 runs to completion in $O(n^3)$ time [9].

Table 3 summarizes the complexity of determining consistency in augmented qualitative networks. Note that when both arc and path consistency are required, we first need to establish path consistency, which results in a complete graph, namely $e = n^2$. Algorithms

[5] Recently, Deville and Van Hentenryck [5] have devised an efficient arc-consistency algorithm which runs in $O(ek)$ time for CPA networks over discrete domains, improving the $O(ek \log k)$ upper bound of AC-3.

	Discrete	Single interval	Multiple intervals
CPA networks	AC $O(ek \log k)$	AC + PC $O(n^3)$	AC + PC $O(n^4K^2)$
PA networks	NP-complete	AC + PC $O(n^3)$	NP-complete
IPA networks	NP-complete	NP-complete	NP-complete

Table 3: Complexity of deciding consistency in augmented qualitative networks.

1. $Q \leftarrow \{(i,k,j) | (i < j), (k \neq i,j)\}$
2. **while** $Q \neq \emptyset$ **do**
3. select and delete any triplet (i,k,j) from Q
4. **if** REVISE$((i,k,j))$ **then**
5. $Q \leftarrow Q \cup$ RELATED-PATHS$((i,k,j))$
6. **end**

Figure 4: PC-2—a path consistency algorithm.

for assembling a solution to augmented qualitative networks are given in the extended version of this paper [10]. Their complexity is bounded by the time needed to decide consistency.

6 Solving General Networks

In this section we focus on solving general networks. First, we describe an exponential brute-force algorithm, and then we investigate the applicability of path consistency algorithms.

Let N be a given network. A *basic label* of arc $i \rightarrow j$, is a selection of a single interval from the interval set (if C_{ij} is quantitative) or a basic relation from the QA element (if C_{ij} is qualitative). A network whose arcs are labeled by basic labels of N is called a *singleton labeling* of N. We may solve N by generating all its singleton labelings, solve each one of them independently, and then combine the results. Specifically, N is consistent if and only if there exists a consistent singleton labeling of N, and the minimal network can be computed by taking the union over the minimal networks of all the singleton labelings.

Each qualitative constraint in a singleton labeling can be translated into a set of up to four linear inequalities on points. For example, a constraint I $\{during\}$ J, can be translated into linear inequalities on the endpoints of I and J, $I^- > J^-$, $I^- < J^+$, $I^+ > J^-$, and $I^+ < J^+$. Using the QUAN translation, these inequalities can be translated into quantitative constraints. It follows, that a singleton labeling is equivalent to an *STP network*—a metric network whose constraints are labeled by single intervals [4]. An STP network can be solved in $O(n^3)$ time [4]; thus, the overall complexity of this decomposition scheme is $O(n^3k^e)$, where n is the number of variables, e is the number of arcs in the constraint graph, and k is the maximum number of basic labels on any arc.

This brute-force enumeration can be pruned significantly by running a backtracking algorithm on a meta-CSP whose variables are the network arcs, and their domains are the possible basic labels. Backtrack assigns a basic label to an arc, as long as the corresponding STP network is consistent and, if no such assignment is possible, it backtracks.

Imposing local consistency among subsets of variables may serve as a preprocessing step to improve backtrack. This strategy has been proven successful (see [3]), as enforcing local consistency can be achieved in polynomial time, while it may substantially reduce the number of dead-ends encountered in the search phase itself. In particular, experimental evaluation shows that enforcing a low consistency level, such as arc or path consistency, gives the best results [3]. Following this rationale, we next show that path consistency, which in general networks amounts to the least amount of preprocessing[6], can be enforced in polynomial time.

To assess the complexity of PC-2 in the context of general networks, we introduce the notion of a *range* of a network [4]. The *range* of a quantitative constraint C, represented by an interval set $\{I_1, \ldots, I_k\}$, where the intervals' extreme points are integers, is $\sup(I_k) - \inf(I_1)$. The range of a *network* is the maximum range over all its quantitative constraints. The range of a network containing rational extreme points is the range of the equivalent integral network, obtained from the input network by multiplying all extreme points by their greatest common divisor. The next theorem shows that the timing of PC-2 is bounded by $O(n^3R^3)$, where R is the range of the network.

Theorem 9 *Let $G = (V, E)$ be a given network. Algorithm PC-2 performs no more than $O(n^3R)$ relaxation steps, and its timing is bounded by $O(n^3R^3)$, where R is the range of G.*

Path consistency can also be regarded as an alternative approach to exhaustive enumeration, serving as an approximation scheme which often yields the minimal network. For example, applying path consistency to the network of Figure 1 produces the minimal network. Although, in general, a path consistent network is not necessarily minimal and may not even be consistent, in some cases path consistency is guaranteed to determine

[6] General networks are trivially arc consistent since unary constraints are represented as binary constraints.

the consistency of a network or to compute the minimal network representation.

Proposition 10 *Let $G = (V, E)$ be a path-consistent network. If the qualitative subnetwork of G is in net(CPA), and the quantitative subnetwork constitutes an STP network, then G is consistent.*

Corollary 11 *Any path-consistent singleton labeling is minimal.*

Note that for networks satisfying the condition of Proposition 10 path consistency is not guaranteed to compute the minimal network (a counterexample is provided in [10]); however, it can be shown that for these networks, the minimal network can be computed using $O(n^2)$ applications of path consistency (see [10]).

We feel that some more temporal problems can be solved by path consistency algorithms; further investigation may reveal new classes for which these algorithms are exact.

7 Conclusions

We described a general network-based model for temporal reasoning capable of handling both qualitative and quantitative information. It facilitates the processing of quantitative constraints on points, and all qualitative constraints between temporal objects. We used constraints satisfaction techniques in solving reasoning tasks in this model. In particular, general networks can be solved by a backtracking algorithm, or by path consistency, which computes an approximation to the minimal network.

Kautz and Ladkin [6] have introduced an alternative model for temporal reasoning. It consists of two components: a metric network and an IA network. These two networks, however, are not connected via internal constraints, rather, they are kept separately, and the inter-component relationships are managed by means of external control. To solve reasoning tasks in this model, Kautz and Ladkin proposed an algorithm which solves each component independently, and then circulates information between the two parts, using the QUAL and QUAN translations, until a fixed point is reached. Our model has two advantages over Kautz and Ladkin's model:

1. It is conceptually clearer, as all information is stored in a single network, and constraint propagation takes place in the knowledge level itself.
2. From computational point of view, we are able to provide tighter bounds for various reasoning tasks. For example, in order to convert a given network into an equivalent path-consistent form, Kautz and Ladkin's algorithm may require $O(n^2)$ informations transferences, resulting in an overall complexity of $O(n^5R^3)$, compared to $O(n^3R^3)$ in our model.

Using our integrated model we were able to identify two new classes of tractable networks. The first class is obtained by augmenting PA and CPA networks with various domain constraints. We showed that some of these networks can be solved using arc and path consistency. The second class consists of networks which can be solved by path consistency algorithms, for example, singleton labelings.

Future research should enrich the representation language to facilitate modeling of more involved reasoning tasks; in particular, we should incorporate non-binary constraints in our model.

Acknowledgments

I would like to thank Rina Dechter and Judea Pearl for providing helpful comments on an earlier draft of this paper.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *CACM*, 26(11):832–843, 1983.

[2] T. L. Dean and D. V. McDermott. Temporal data base management. *Artificial Intelligence*, 32:1–55, 1987.

[3] R. Dechter and I. Meiri. Experimental evaluation of preprocessing techniques in constraint satisfaction problems. In *Proceedings of IJCAI-89*, pages 271–277, Detroit, MI, 1989.

[4] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49, 1991.

[5] Y. Deville and P. Van Hentenryck. An efficient arc consistency algorithm for a class of CSP problems. In *Proceedings of IJCAI-91*, Sydney, Australia, 1991.

[6] H. Kautz and P. B. Ladkin. Integrating metric and qualitative temporal reasoning. In *Proceedings of AAAI-91*, Anaheim, CA, 1991.

[7] P. B. Ladkin and R. D. Maddux. On binary constraint networks. Technical report, Kestrel Institute, Palo Alto, CA, 1989.

[8] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977.

[9] A. K. Mackworth and E. C. Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, 25(1):65–74, 1985.

[10] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning. Technical Report TR-160, UCLA Cognitive Systems Laboratory, Los Angeles, CA, 1991.

[11] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, (7):95–132, 1974.

[12] P. van Beek. *Exact and Approximate Reasoning about Qualitative Temporal Relations*. PhD thesis, University of Waterloo, Ontario, Canada, 1990.

[13] P. van Beek. Reasoning about qualitative temporal information. In *Proceedings of AAAI-90*, pages 728–734, Boston, MA, 1990.

[14] M. Vilain and H. Kautz. Constraint propagation algorithms for temporal reasoning. In *Proceedings of AAAI-86*, pages 377–382, Philadelphia, PA, 1986.